Rafał Schmidt

Unlocking the Complexity of Documentation Creation: Perspectives, Challenges, and Criticality in a Large Organization

allegro

Rafał Schmidt

Unlocking the Complexity of Documentation Creation: Perspectives, Challenges, and Criticality in a Large Organization

I've never heard (from devs)

++ 2 13 14 8 15

29.8k

++ 2 13 14 8 15

29.8k

The documentation is super intuitive

++ 2 13 14 8 15

29.8k

−−

The documentation is super intuitive, we do not need code examples for our api.


EASTMAN
DELI FOODS
PROPER TASTY
OOKED HAM
9
SLICES
25 FEB

Looks like bad documentation to me.

++ 1.1k −− Share •••

?!

New solution?

Features?

Features?

Restrictions?

Features?

Restrictions?

Alternatives?

Features?

Restrictions?

Alternatives?

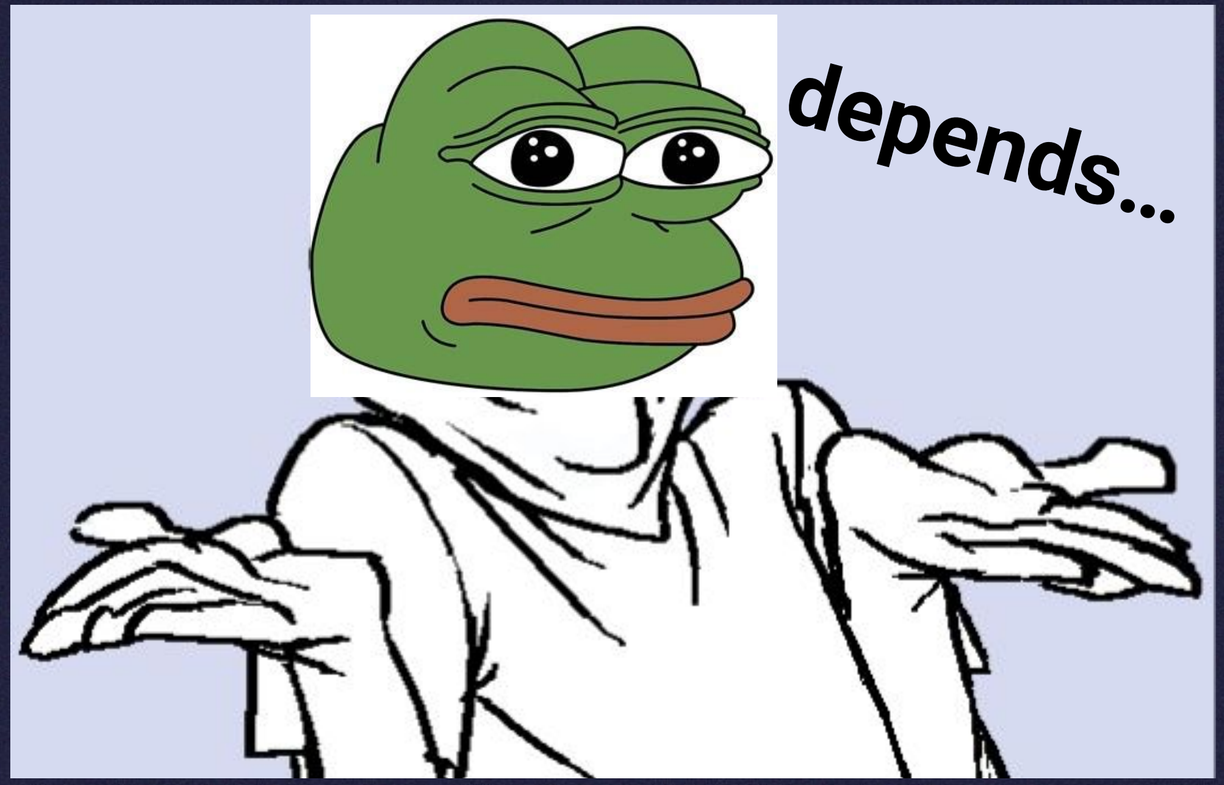
Credibility?

Features?

Restrictions?

Alternatives?

Credibility?



Well written?

First impression

Unclear

++ 3.0k −− Reply Share •••

"Code is like a joke.

++ 3.0k −− Reply Share •••

"Code is like a joke. If you have to explain it, then it's bad."

++ 3.0k −− Reply Share •••

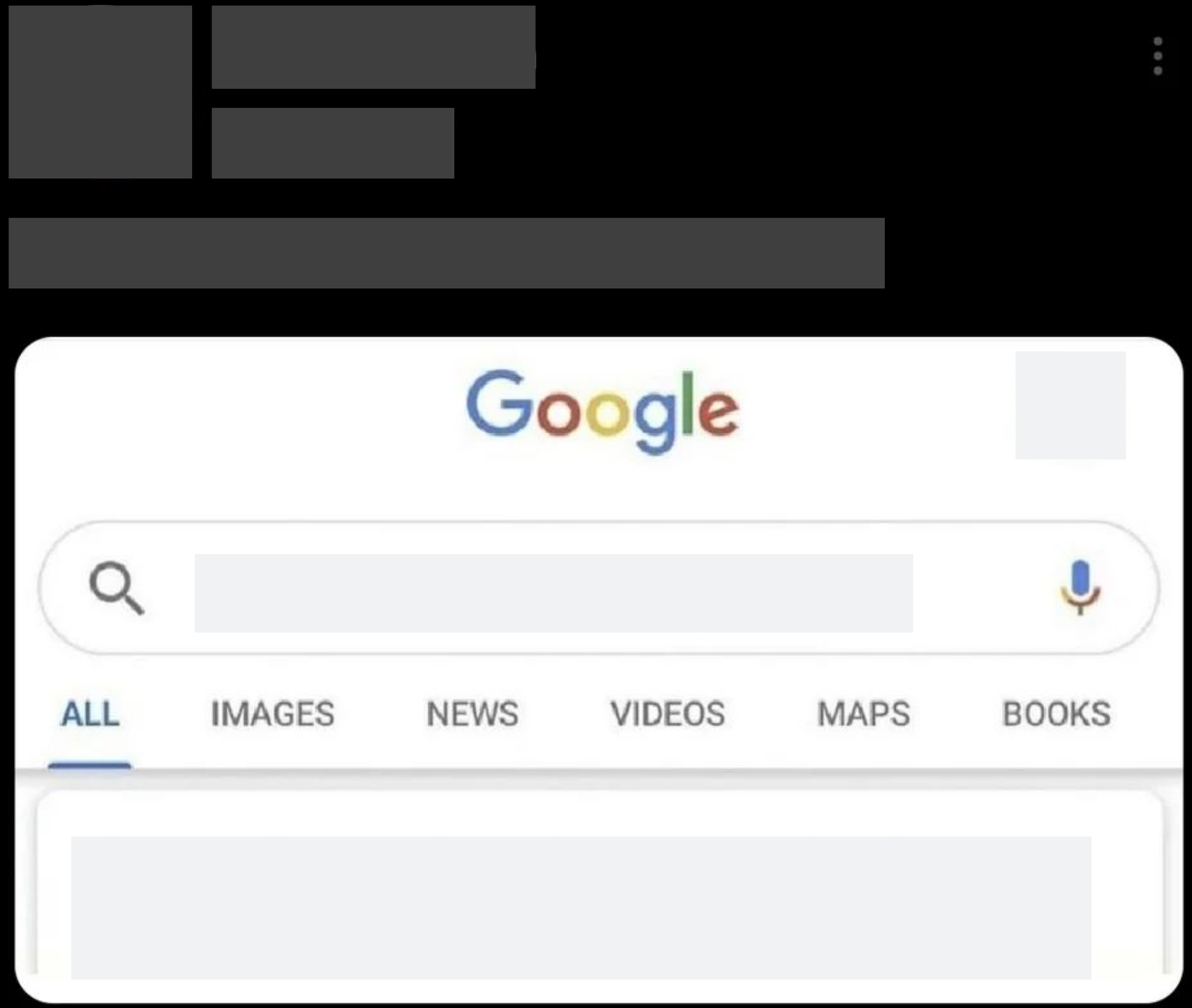
"Code is like a joke. If you have to explain it, then it's bad."

~ Coworker whose code was truly a joke.

++ 3.0k −− Reply Share •••

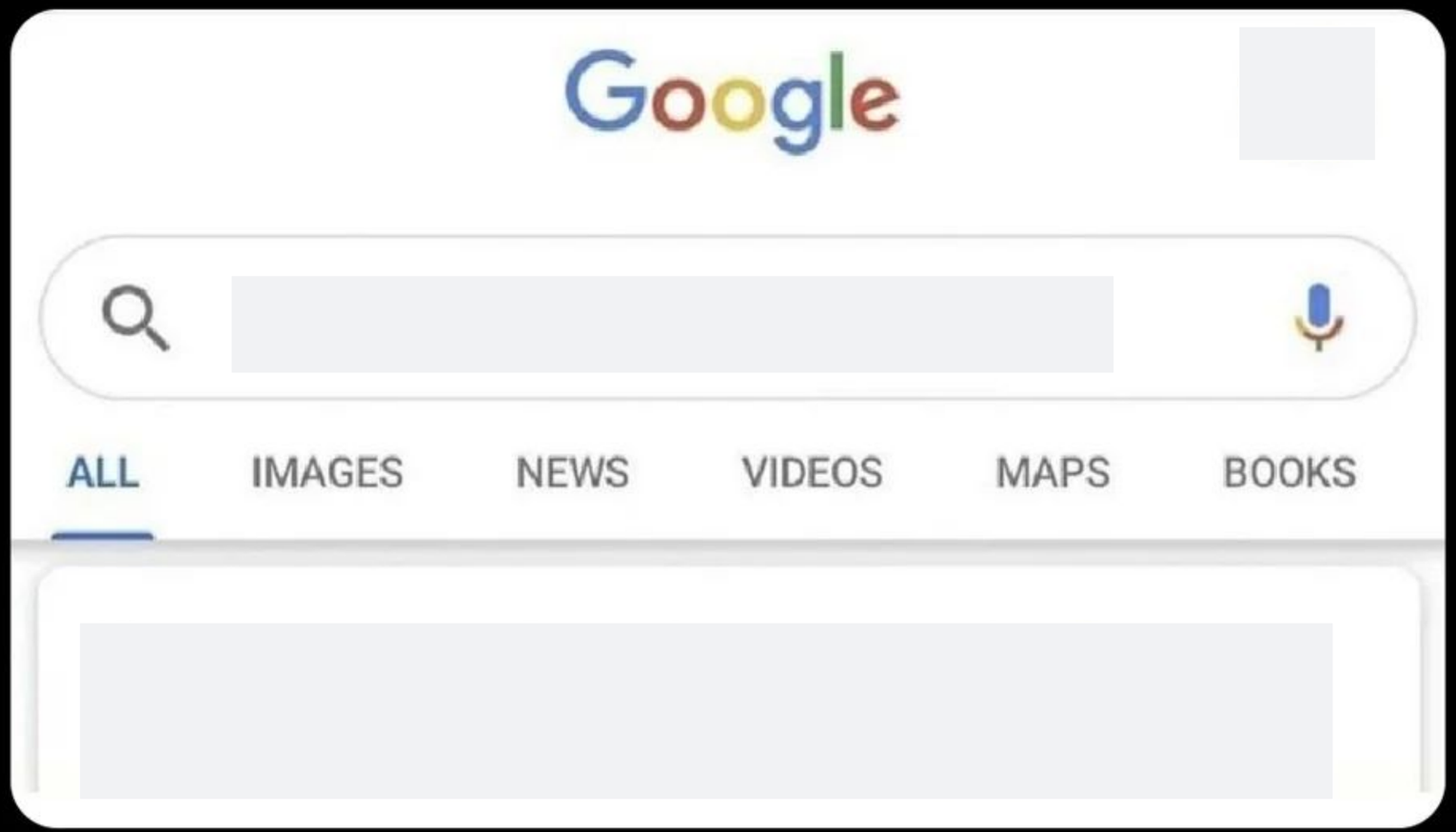
Unclear

Null


Google
ALL
IMAGES
NEWS
VIDEOS
MAPS
BOOKS

[redacted]

software documentation be like



[redacted]

software documentation be like

[redacted]

software documentation be like

"How do I cure a hangover?"

> Drink water before bed, drink less alcohol

++ 438 −− Share •••

How do I prevent hunger

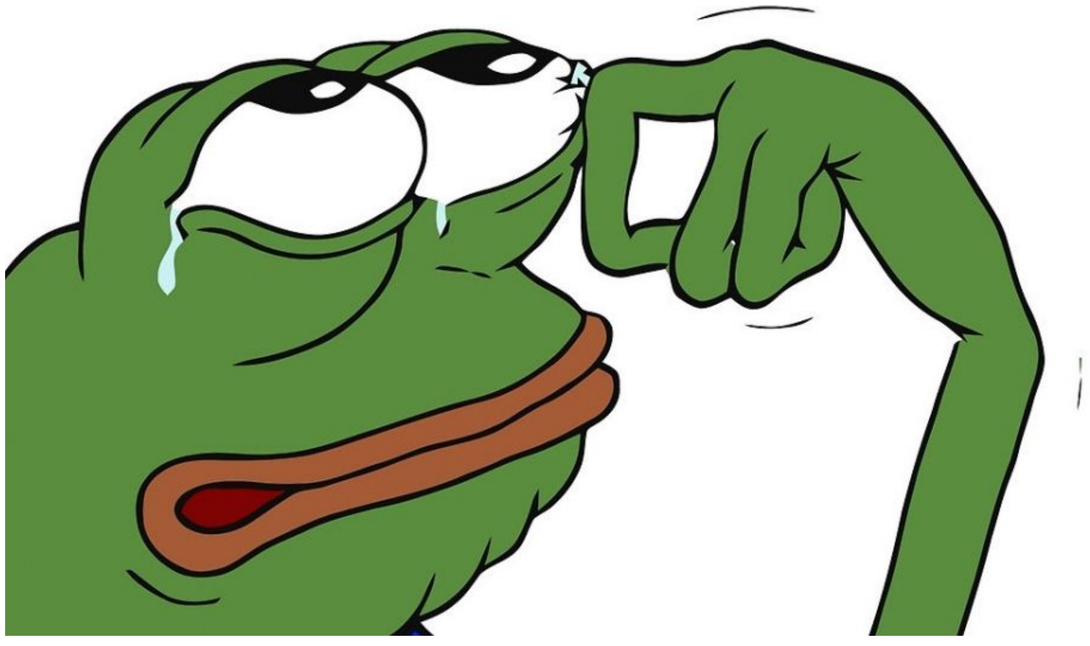
Eat food

++ 8 −− Share ···

How to cure food poisoning?

> Don't eat poisonous food

++ 243 −− Share ···



Grr. Who writes this stuff?

++ 708 −− Share •••

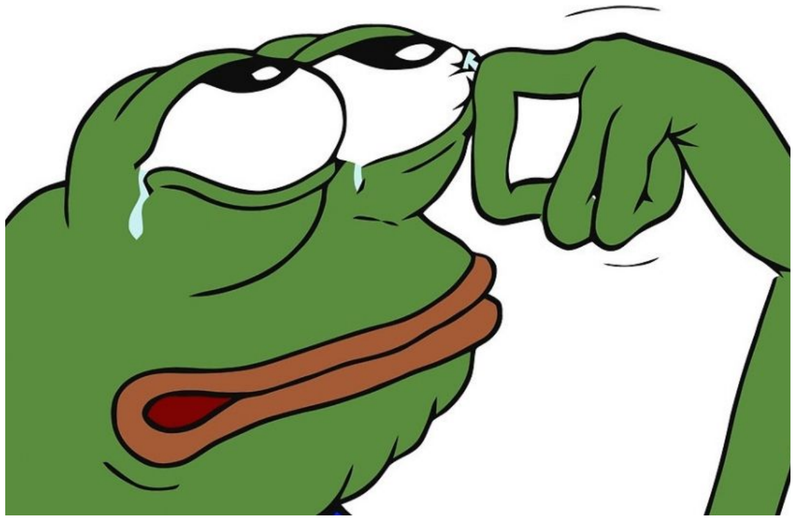
Grr. Who writes this stuff? I want to meet them and tell them a few things.

++ 708 −− Share •••

Unclear

Null

Obvious



Lesson?

It's simple..

++ 221 −− Share •••

It's simple.. writing good documentation is a is a non trivial skill

++ 221 −− Share •••

It's simple.. writing good documentation is a is a non trivial skill that most devs aren't good at.

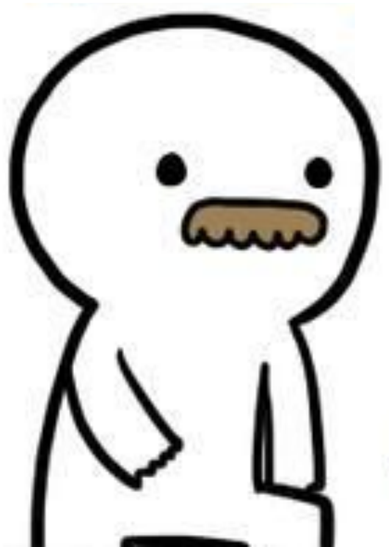
It's simple.. writing good documentation is a is a non trivial skill that most devs aren't good at. Mainly because they suffer from the curse of knowledge about the system they are trying to document.

> It's simple.. writing good documentation is a is a non trivial skill that most devs aren't good at. Mainly because they suffer from the curse of knowledge about the system they are trying to document.
>
> ++ 221 −− Share •••

Array.prototype.reduce()

Wrap up…

?



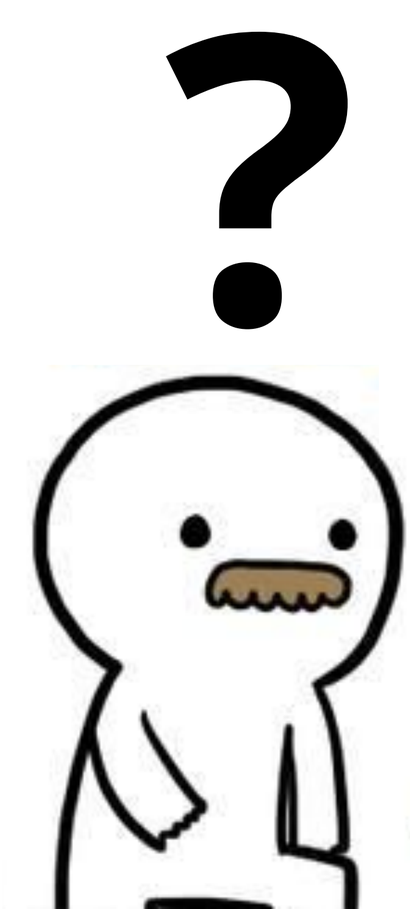

Przykłady użycia

Rate your tools - Development Ecosystem feedback wanted

Hello,

I'm done with the report and here is the brief TL;DR summary:

b. lack of documentation making us wandering in the search for knowledge,



Goal

Goal

1. Target groups - “who”

Goal

1. Target groups - “who”
2. Use cases - “what”

Goal

1. Target groups - “who”
2. Use cases - “what”
3. Tooling - “how”

Why?


My friend told the lady at McDonalds
"It's Stephen, with a ph"...
216
Thank You For Choosing


My friend told the lady at McDonalds
"It's Stephen, with a ph"...
Phteven
216
Thank You For Choosing


My friend told the lady at McDonalds
"It's Stephen, with a ph"...
Phteven
216
Thank You For Choosing


PHTEVEN

Target groups - “who”

++ 1.1k −− Reply Share •••

I comment so that

I comment so that *I'll* be able to remember what the hell I was thinking 6 months ago

++ 1.1k −− Reply Share •••


SPACE RANGER
LIGHTYEAR

New hires

++ 503 −− Share ···

Exactly. New hires and junior developers represent

New hires and junior developers represent a golden opportunity to identify cargo cult policies, tribal knowledge, and absent or incorrect documentation in your product.

Bulletproof?

++ 9 −− Share ···

So hard to make foolproof documentation.

So hard to make foolproof documentation. Fools are too clever.

Loop?

much like writting code, documentation also should go iterative process.

much like writting code, documentation also should go iterative process. If there is someone complaining, i don't understand the doc, ask them back which part that need clarification.

much like writting code, documentation also should go iterative process. If there is someone complaining, i don't understand the doc, ask them back which part that need clarification.

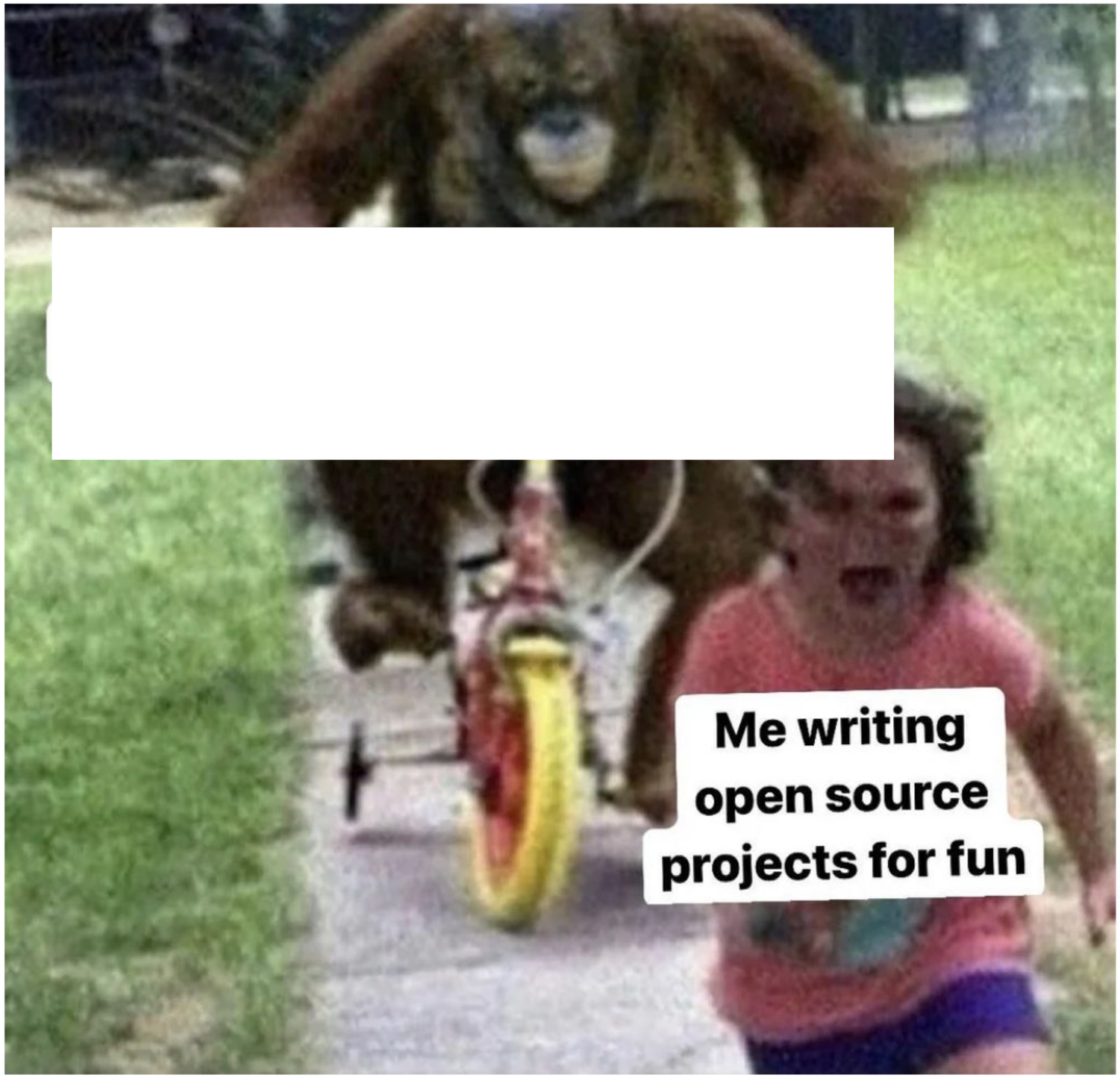
"The example is vague and not explaining the use case". Okay, then write the doc again with common use case. If the doc grow excessively long, split the doc into intuitive section.

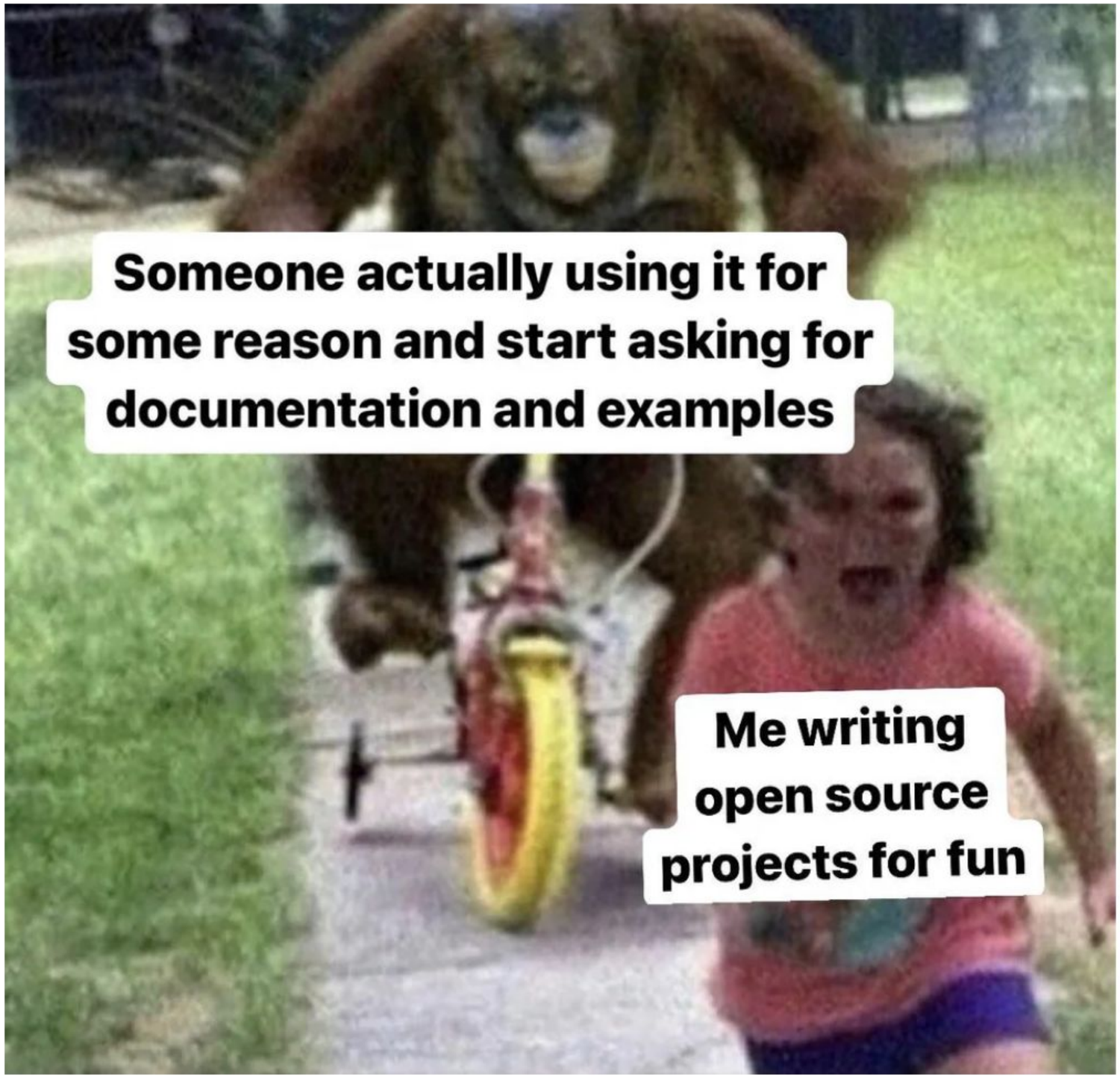
much like writting code, documentation also should go iterative process. If there is someone complaining, i don't understand the doc, ask them back which part that need clarification.

"The example is vague and not explaining the use case". Okay, then write the doc again with common use case. If the doc grow excessively long, split the doc into intuitive section.

Loop

Responsibility?


Me writing
open source
projects for fun


Someone actually using it for some reason and start asking for documentation and examples
Me writing open source projects for fun

Target groups - “who”

Target groups - “who”

You

Target groups - “who”

You
Team devs

Target groups - “who”

- You
- Team devs
- Testers

Target groups - “who”

You
Team devs
Testers
Other devs

Target groups - “who”

You
Team devs
Testers
Other devs
Principals

Target groups - “who”

You
Team devs
Testers
Other devs
Principals

Analysts

Target groups - “who”

You
Team devs
Testers
Other devs
Principals

Analysts
Business

Target groups - “who”

You
Team devs
Testers
Other devs
Principals

Analysts
Business
On duty

Target groups - “who”

You
Team devs
Testers
Other devs
Principals

Analysts
Business
On duty
Compliance

Target groups - “who”

You
Team devs
Testers
Other devs
Principals

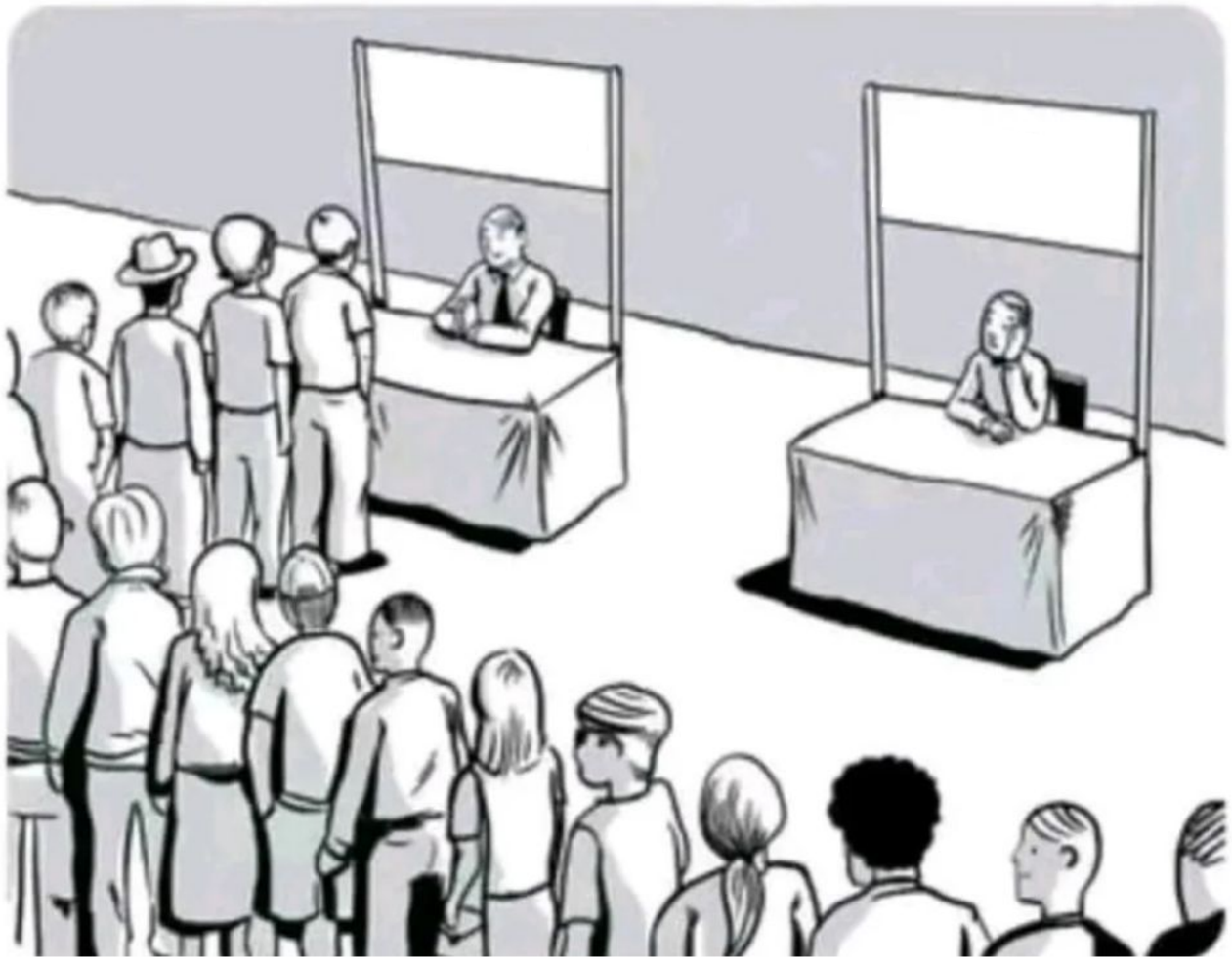
Analysts
Business
On duty
Compliance
Auditors

Use cases - “what”

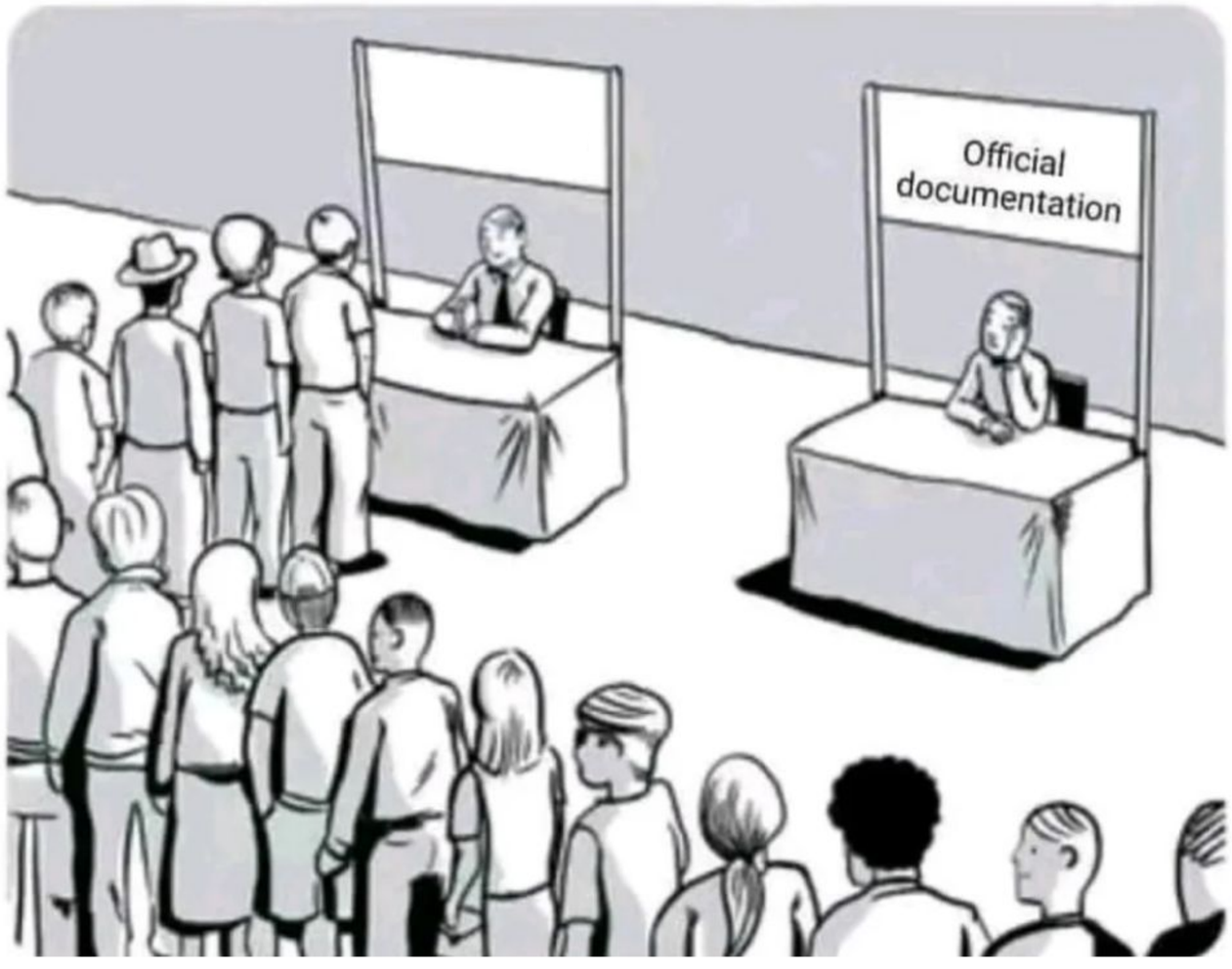
(mixed with recommendations)

Developers searching for information

Developers searching for information

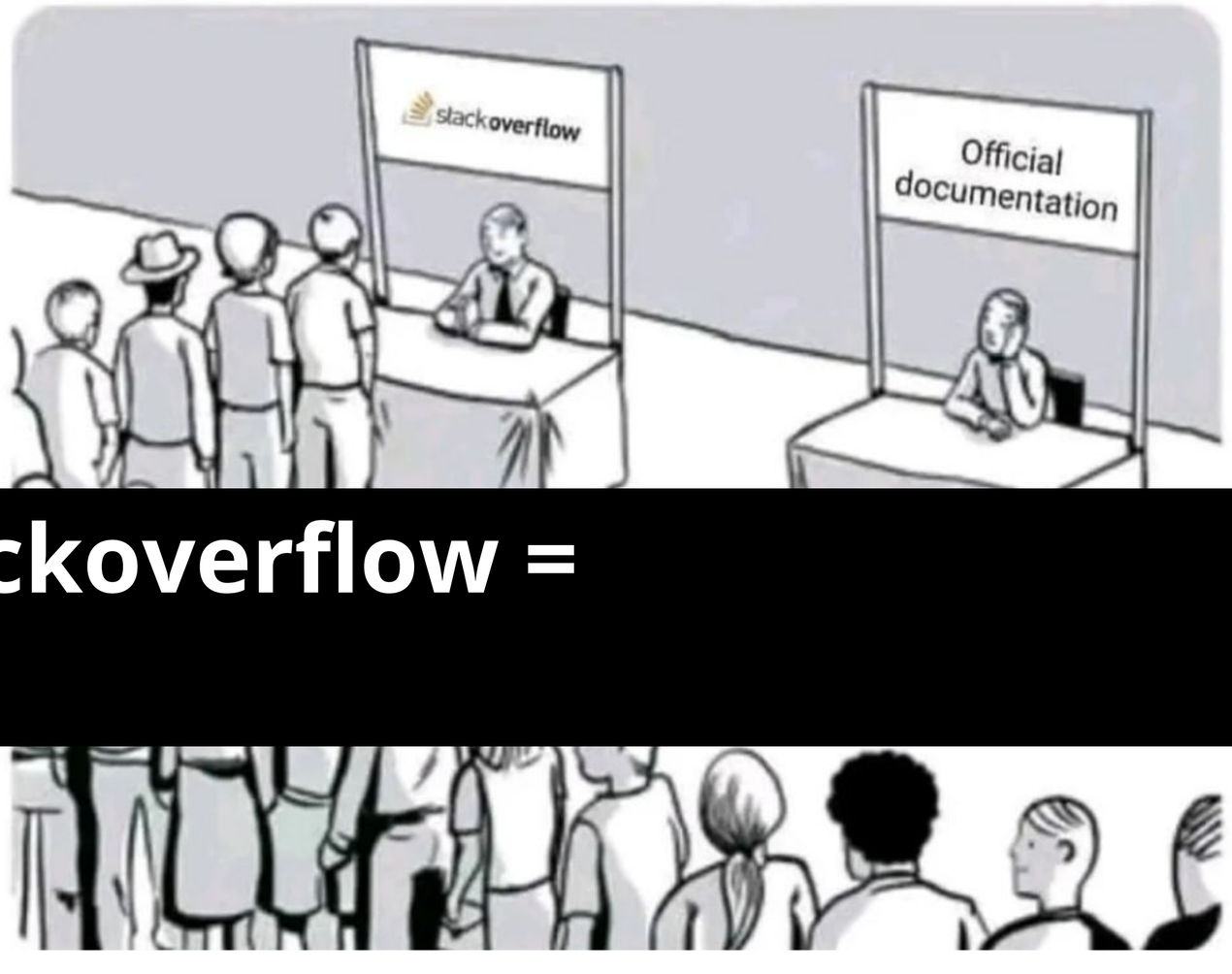


Developers searching for information



Developers searching for information

Developers searching for information



stackoverflow =

Developers searching for information

Developers searching for information

stackoverflow = target_group
*** use_cases**
*** living_doc_ratio**

Developers searching for information

problem?

Developers searching for information

Developers searching for information

Developers searching for information

The only thing I miss in some documentations is examples

++ 148 −− Reply Share •••

> The only thing I miss in some documentations is examples, they just define, this thing does this and done.

++ 148 −− Reply Share •••

> The only thing I miss in some documentations is examples, they just define, this thing does this and done. Sometimes I don't understand how to use them actually

++ 148 −− Reply Share •••

Syntax Highlighting

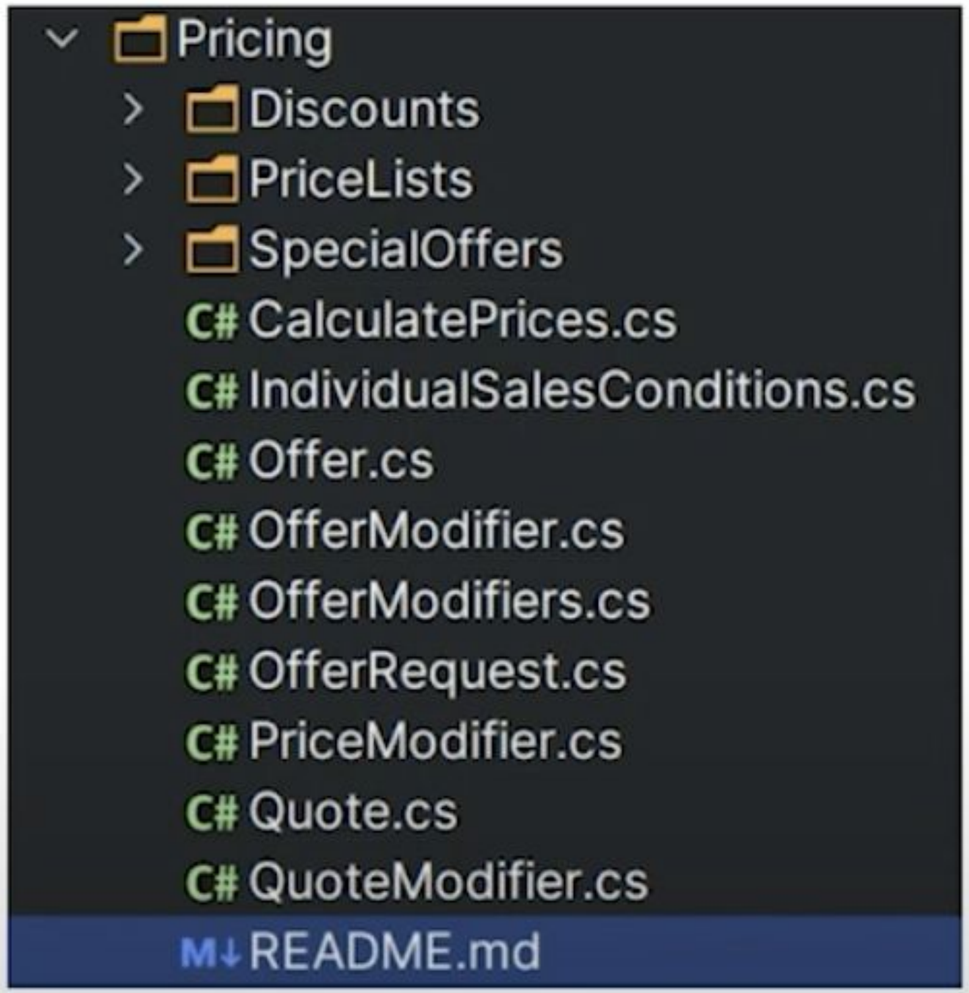
````
```json
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25
}
```
````

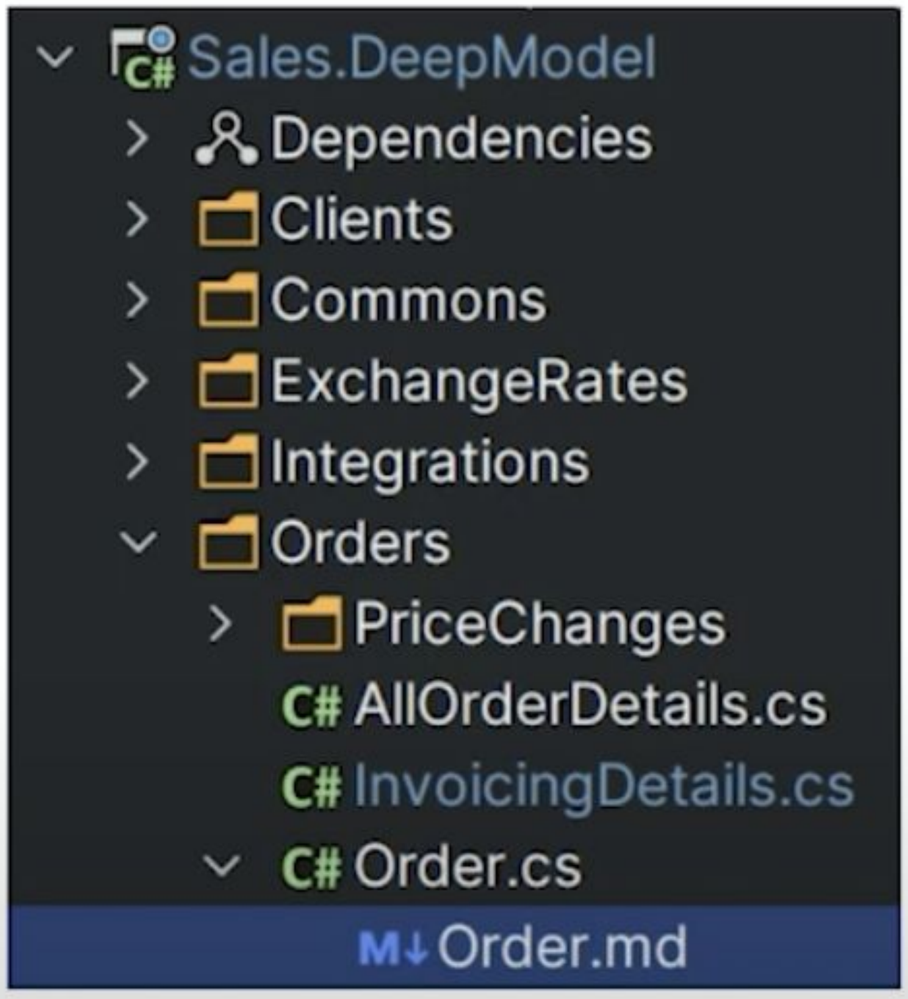
Syntax Highlighting

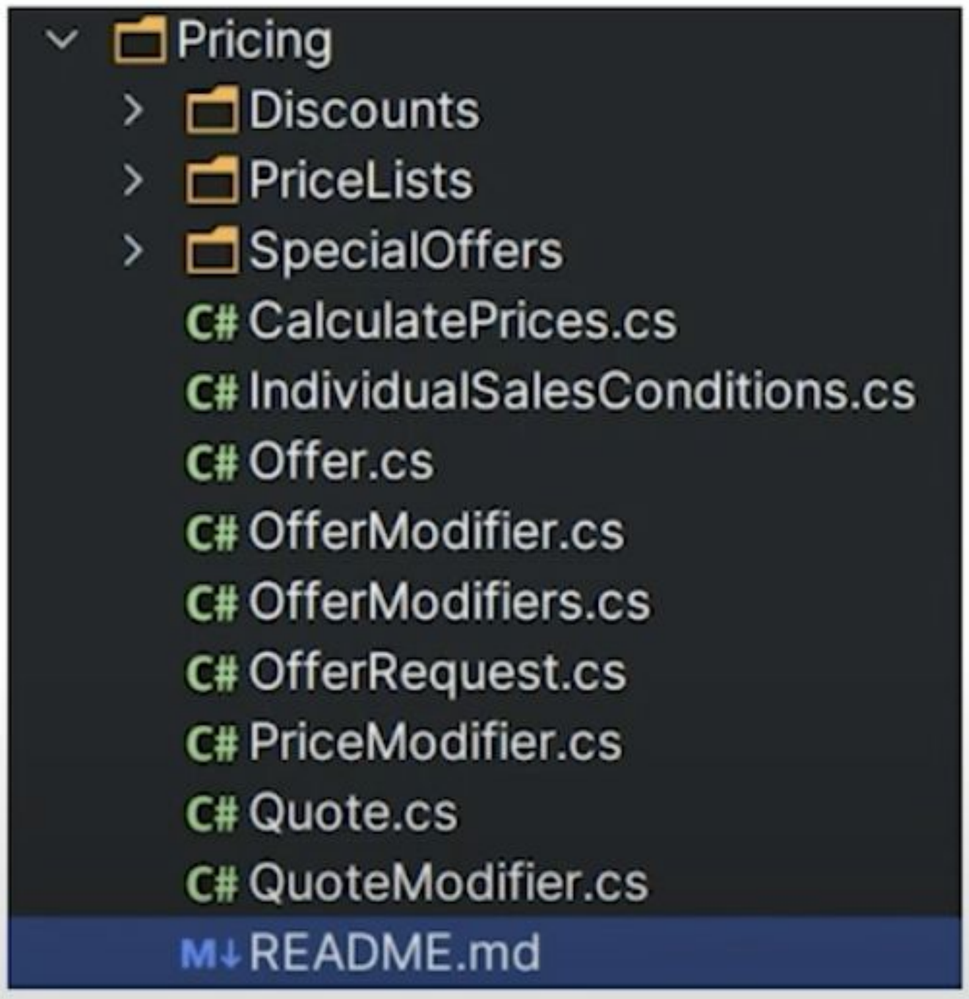
````
```json
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25
}
```
````

The rendered output looks like this:

```json
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25
}
```


Pricing
Discounts
PriceLists
SpecialOffers
C# CalculatePrices.cs
C# IndividualSalesConditions.cs
C# Offer.cs
C# OfferModifier.cs
C# OfferModifiers.cs
C# OfferRequest.cs
C# PriceModifier.cs
C# Quote.cs
C# QuoteModifier.cs
M↓ README.md


Sales.DeepModel
Dependencies
Clients
Commons
ExchangeRates
Integrations
Orders
PriceChanges
AllOrderDetails.cs
InvoicingDetails.cs
Order.cs
Order.md


Pricing
Discounts
PriceLists
SpecialOffers
CalculatePrices.cs
IndividualSalesConditions.cs
Offer.cs
OfferModifier.cs
OfferModifiers.cs
OfferRequest.cs
PriceModifier.cs
Quote.cs
QuoteModifier.cs
README.md

Automatic?

```yaml
openapi: "3.0.0"
info:
  version: 1.0.0
  title: Swagger Petstore
  license:
    name: MIT
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags:
        - pets
      parameters:
        - name: limit
          in: query
          description: How many items to return at one
            time (max 100)
          required: false
          schema:
            type: integer
            format: int32
    responses:
      '200':
        description: A paged array of pets
        headers:
          x-next:
            description: A link to the next page of
              responses
            schema:
              type: string
```

Swagger Petstore 1.0.0 OAS3

MIT

Servers

http://petstore.swagger.io/v1

pets

GET `/pets` List all pets

POST `/pets` Create a pet

GET `/pets/{petId}` Info for a specific pet

Models

examples?

Generating Swagger example requests with Swashbuckle

Generating Swagger example requests with Swashbuckle

```
public class DeliveryOptionsSearchModelExample
{
    public object GetExamples()
    {
        return new DeliveryOptionsSearchModel
        {
            Lang = "en-GB",
            Currency = "GBP",
            Address = new AddressModel
            {
                Address1 = "1 Gwalior Road",
                Locality = "London",
                Country = "GB",
                PostalCode = "SW15 1NP"
            },
            Items = new[]
            {
                new ItemModel
                {
                    ItemId = "ABCD",
                    ItemType = ItemType.Product,
                    Price = 20,
                    Quantity = 1,
                    RestrictedCountries = new[] { "US" }
                }
            }
        };
    }
```

Generating Swagger example requests with Swashbuckle

```
public class DeliveryOptionsSearchModelExample : IExamplesProvider
{
    public object GetExamples()
    {
        return new DeliveryOptionsSearchModel
        {
            Lang = "en-GB",
            Currency = "GBP",
            Address = new AddressModel
            {
                Address1 = "1 Gwalior Road",
                Locality = "London",
                Country = "GB",
                PostalCode = "SW15 1NP"
            },
            Items = new[]
            {
                new ItemModel
                {
                    ItemId = "ABCD",
                    ItemType = ItemType.Product,
                    Price = 20,
                    Quantity = 1,
                    RestrictedCountries = new[] { "US" }
                }
            }
        };
    }
```

Generating Swagger example requests with Swashbuckle

```
public class DeliveryOptionsSearchModelExample : IExamplesProvider
{
    public object GetExamples()
    {
        return new DeliveryOptionsSearchModel
        {
            Lang = "en-GB",
            Currency = "GBP",
            Address = new AddressModel
            {
                Address1 = "1 Gwalior Road",
                Locality = "London",
                Country = "GB",
                PostalCode = "SW15 1NP"
            },
            Items = new[]
            {
                new ItemModel
                {
                    ItemId = "ABCD",
                    ItemType = ItemType.Product,
                    Price = 20,
                    Quantity = 1,
                    RestrictedCountries = new[] { "US" }
                }
            }
        };
    }
```

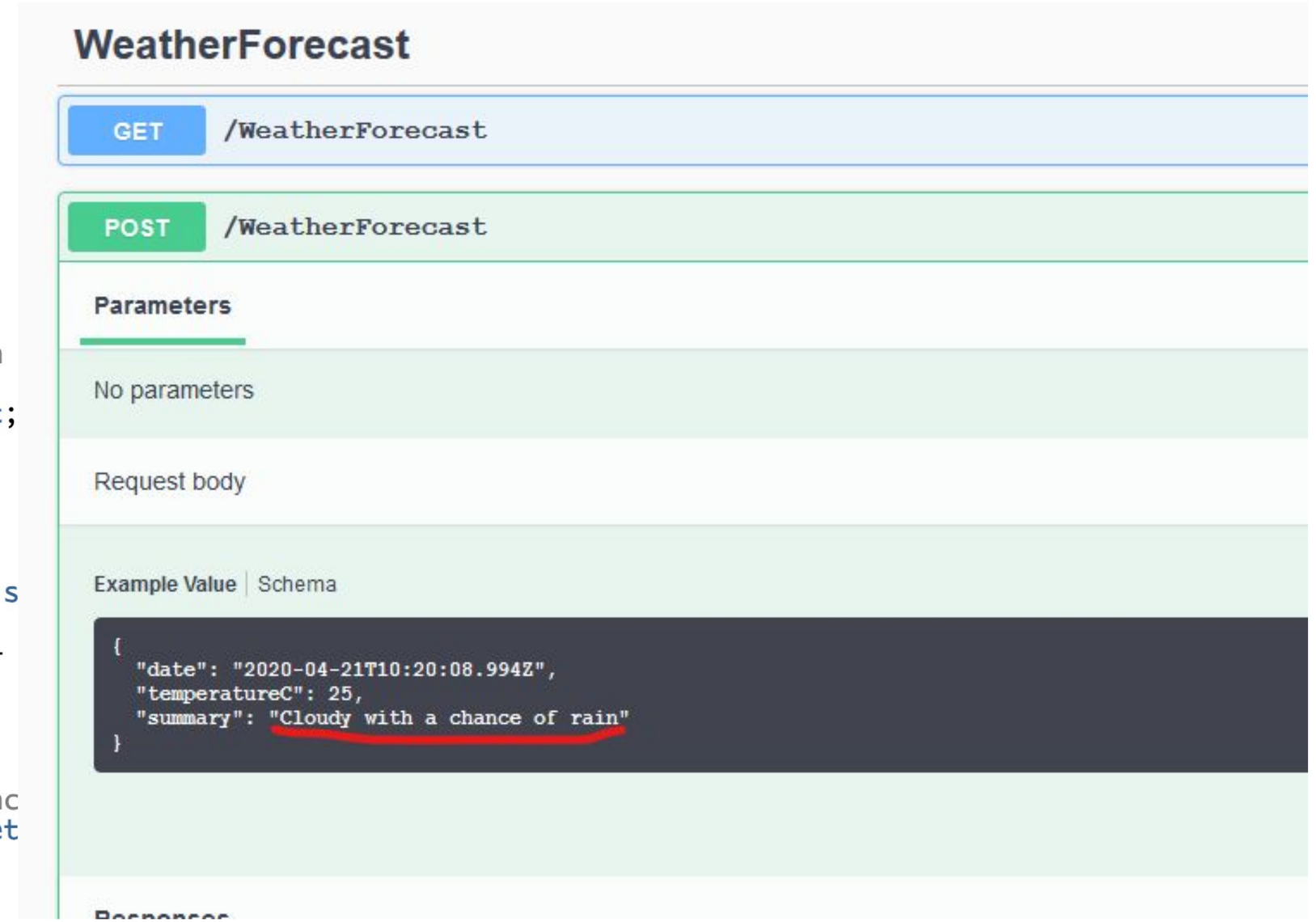
```
services.AddSwaggerGen(c =>
   {
        c.SwaggerDoc("v1", new Info { Title = "My API",
        c.OperationFilter<ExamplesOperationFilter>();
```

Generating Swagger example requests with Swashbuckle

```
11
12  public class WeatherForecast
13  {
14      /// <summary>
15      /// The date of the forecast in ISO-whatever format
16      /// </summary>
17      public DateTime Date { get; set; }
18
19      /// <summary>
20      /// Temperature in celcius
21      /// </summary>
22      /// <example>25</example>
23      public int TemperatureC { get; set; }
24
25      public int TemperatureF => 32 + (int)(TemperatureC / 0.5556);
26
27      /// <summary>
28      /// A textual summary
29      /// </summary>
30      /// <example>Cloudy with a chance of rain</example>
31      public string Summary { get; set; }
32  }
```

Generating Swagger example requests with Swashbuckle

```
11
12  public class WeatherForecast
13  {
14      /// <summary>
15      /// The date of the forecast in
16      /// </summary>
17      public DateTime Date { get; set;
18
19      /// <summary>
20      /// Temperature in celcius
21      /// </summary>
22      /// <example>25</example>
23      publi int TemperatureC { get; s
24
25      publ c int TemperatureF => 32 +
26
27      /// <summary>
28      /// A textual summary
29      /// </summary>
30      /// <example>Cloudy with a chanc
31      public string Summary { get; set
32  }
```



self-documenting code sound so good in theory,

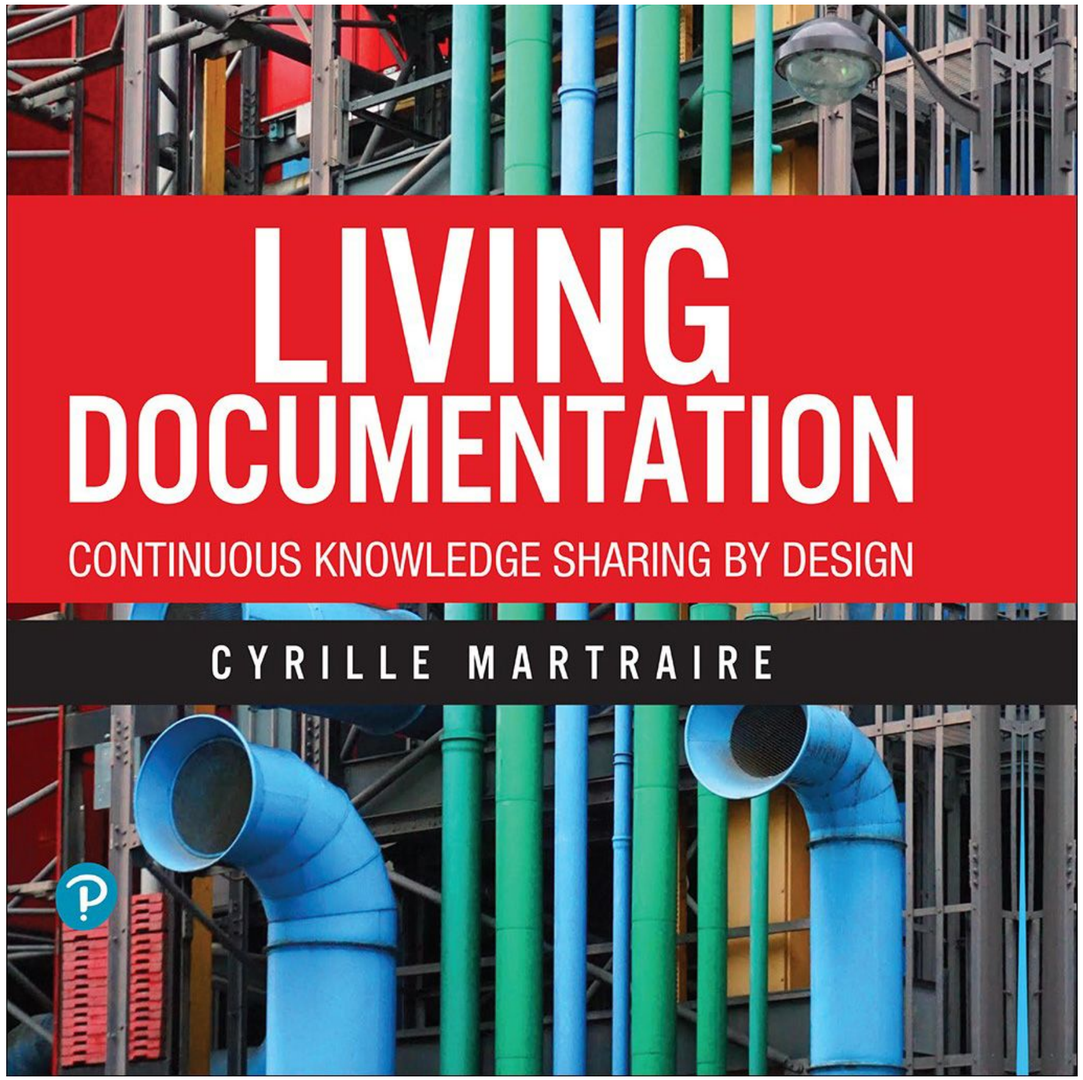
++ 426 -- Reply Share •••

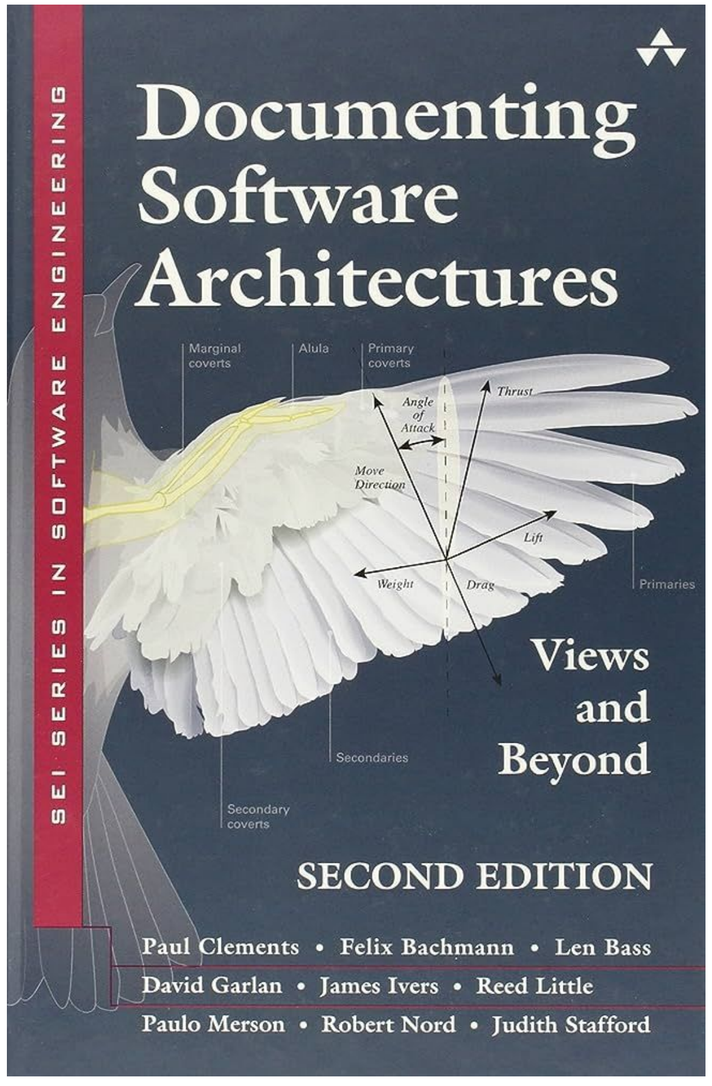
> self-documenting code sound so good in theory, but in practice, that shit just works when the project is small, and little people works on it.

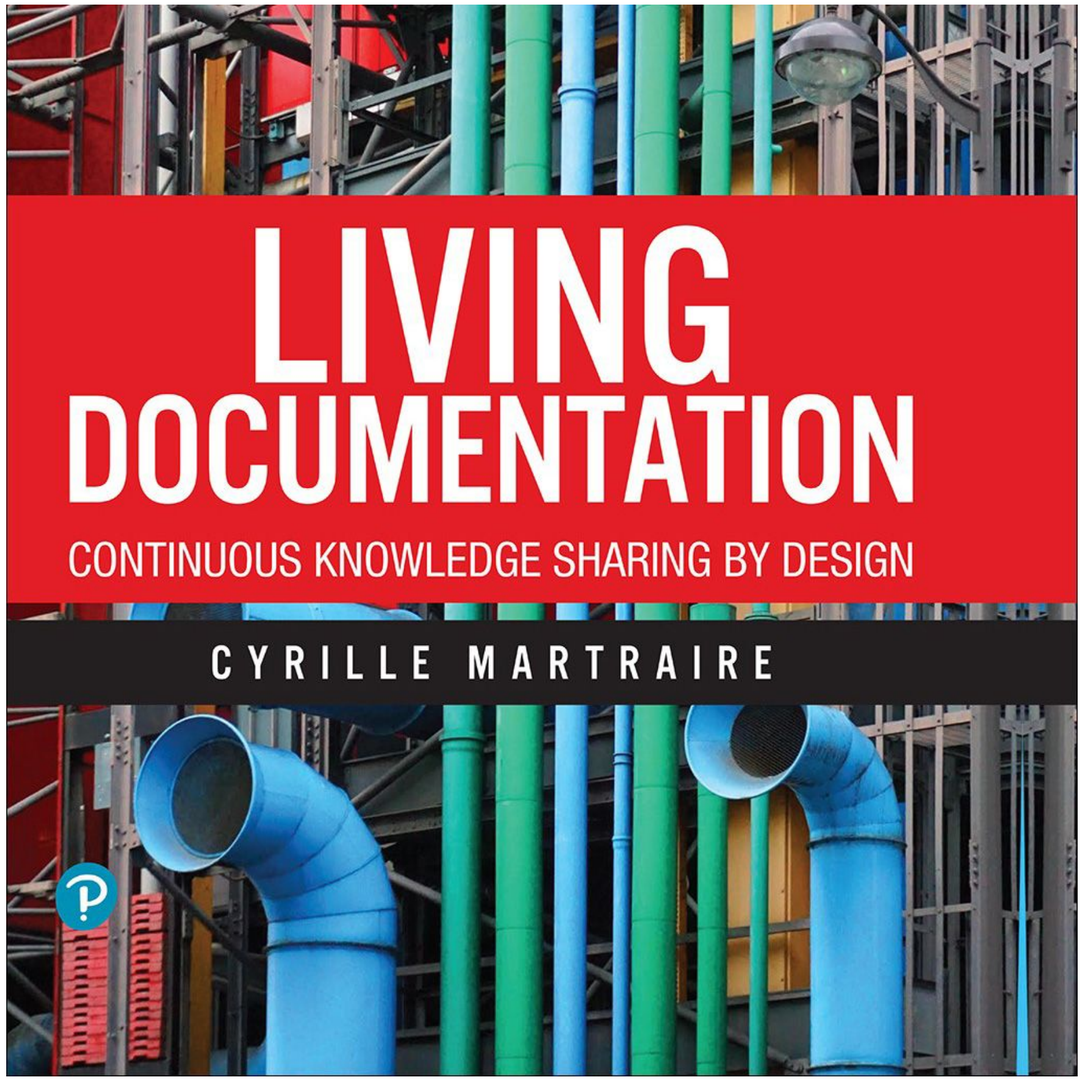
++ 426 −− Reply Share •••

```
///<summary>
///   Gets or sets the widget.
///</summary>
public object Widget { get; set; }
```

++ 585 −− Share ···


LIVING
DOCUMENTATION
CONTINUOUS KNOWLEDGE SHARING BY DESIGN
CYRILLE MARTRAIRE

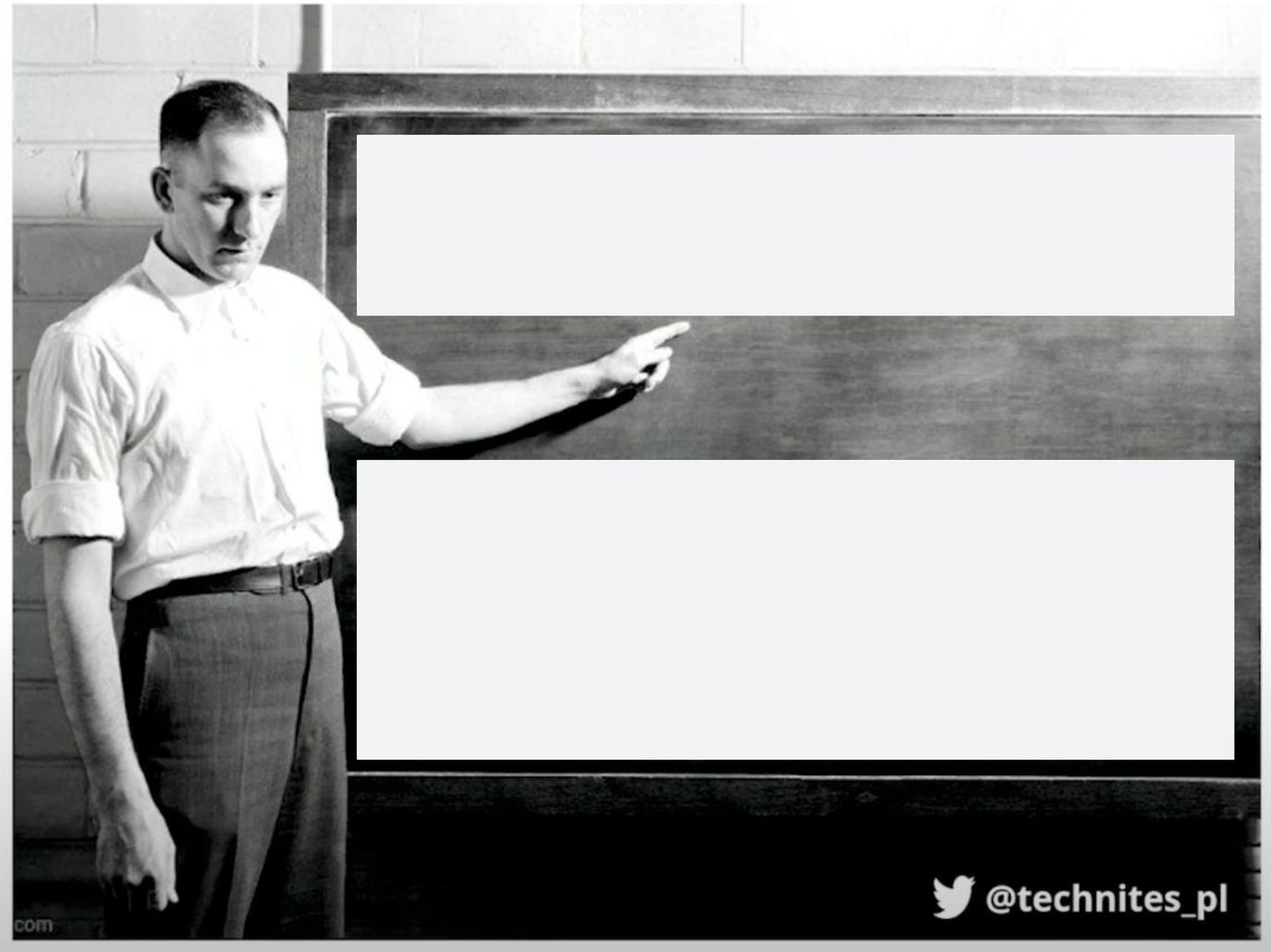

SEI SERIES IN SOFTWARE ENGINEERING
Documenting Software Architectures
Marginal coverts
Alula
Primary coverts
Angle of Attack
Thrust
Move Direction
Lift
Weight
Drag
Primaries
Secondaries
Secondary coverts
Views and Beyond
SECOND EDITION
Paul Clements • Felix Bachmann • Len Bass
David Garlan • James Ivers • Reed Little
Paulo Merson • Robert Nord • Judith Stafford


LIVING DOCUMENTATION
CONTINUOUS KNOWLEDGE SHARING BY DESIGN
CYRILLE MARTRAIRE

BoilingFrogs

DOKUMENTACJA, KTÓRA SAMA SIĘ PISZE

MARCIN MARKOWSKI
Software Architect and Trainer
TECHNITES_PL



Boiling Frogs

DOKUMENTACJA, KTÓRA SAMA SIĘ PISZE

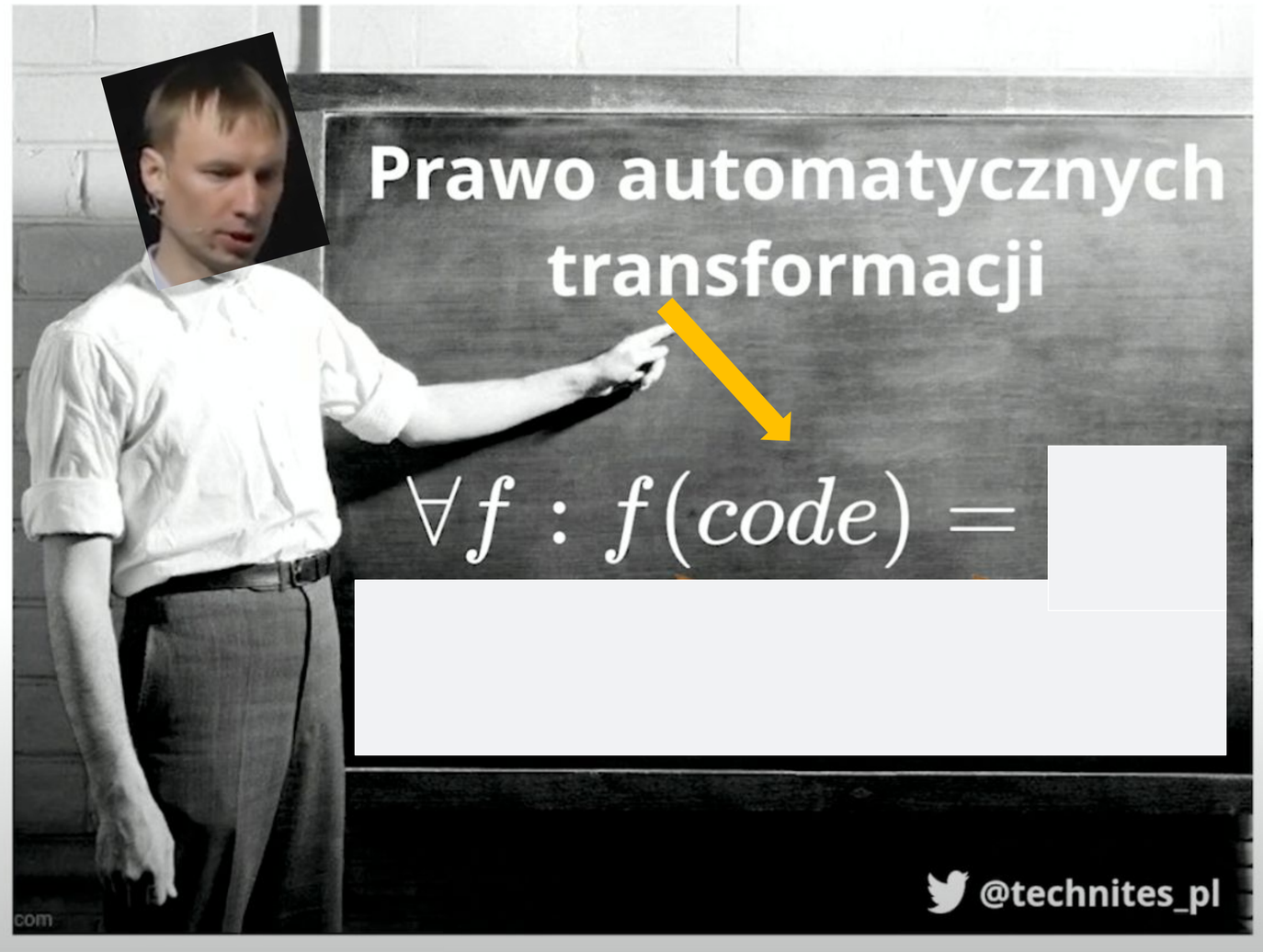
MARCIN MARKOWSKI
Software Architect and Trainer
TECHNITES_PL

17:49 / 55:49

BoilingFrogs

DOKUMENTACJA, KTÓRA SAMA SIĘ PISZE

MARCIN MARKOWSKI
Software Architect and Trainer
TECHNITES_PL

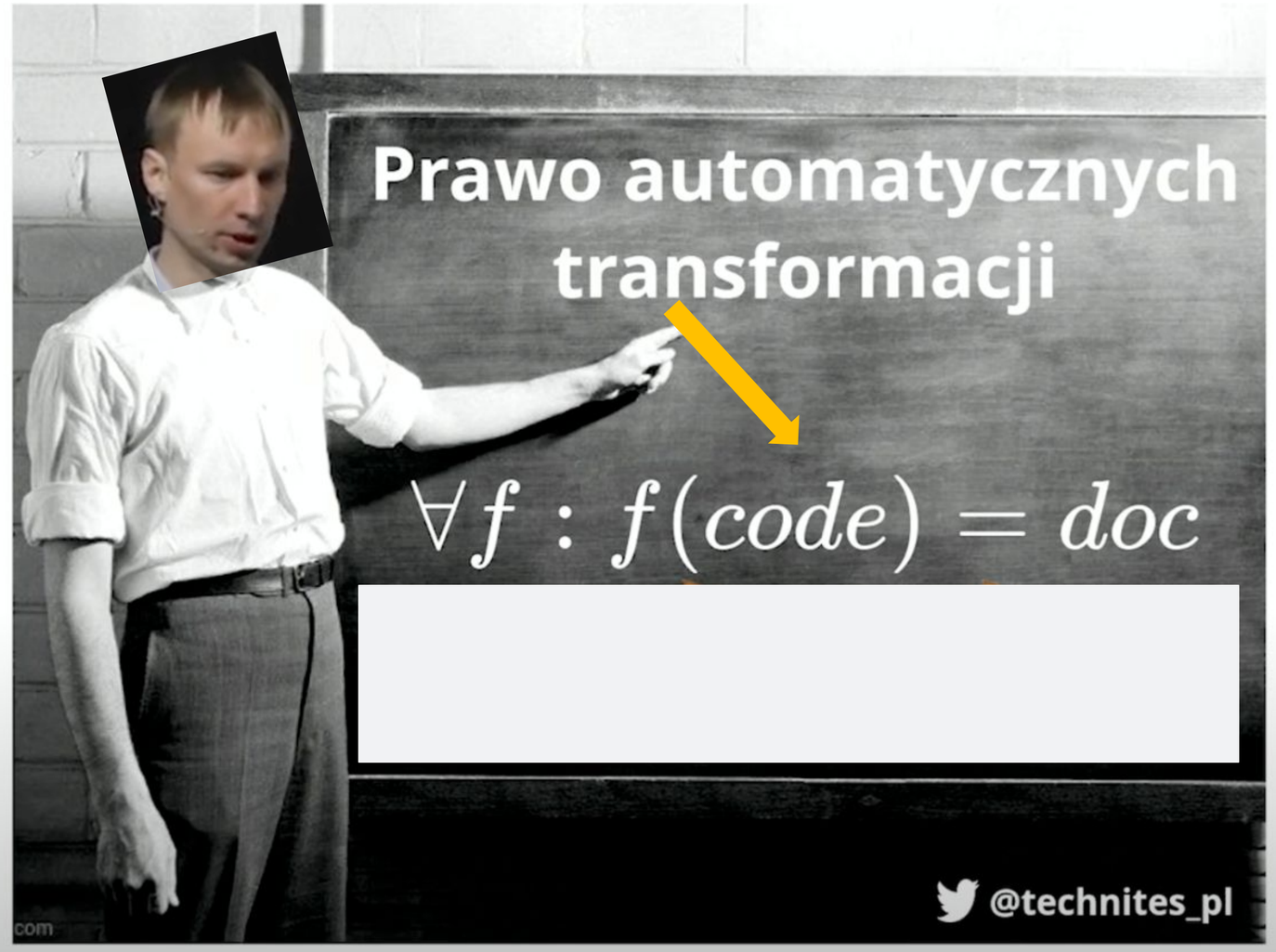


17:49 / 55:49

BoilingFrogs

DOKUMENTACJA, KTÓRA SAMA SIĘ PISZE

MARCIN MARKOWSKI
Software Architect and Trainer
TECHNITES_PL

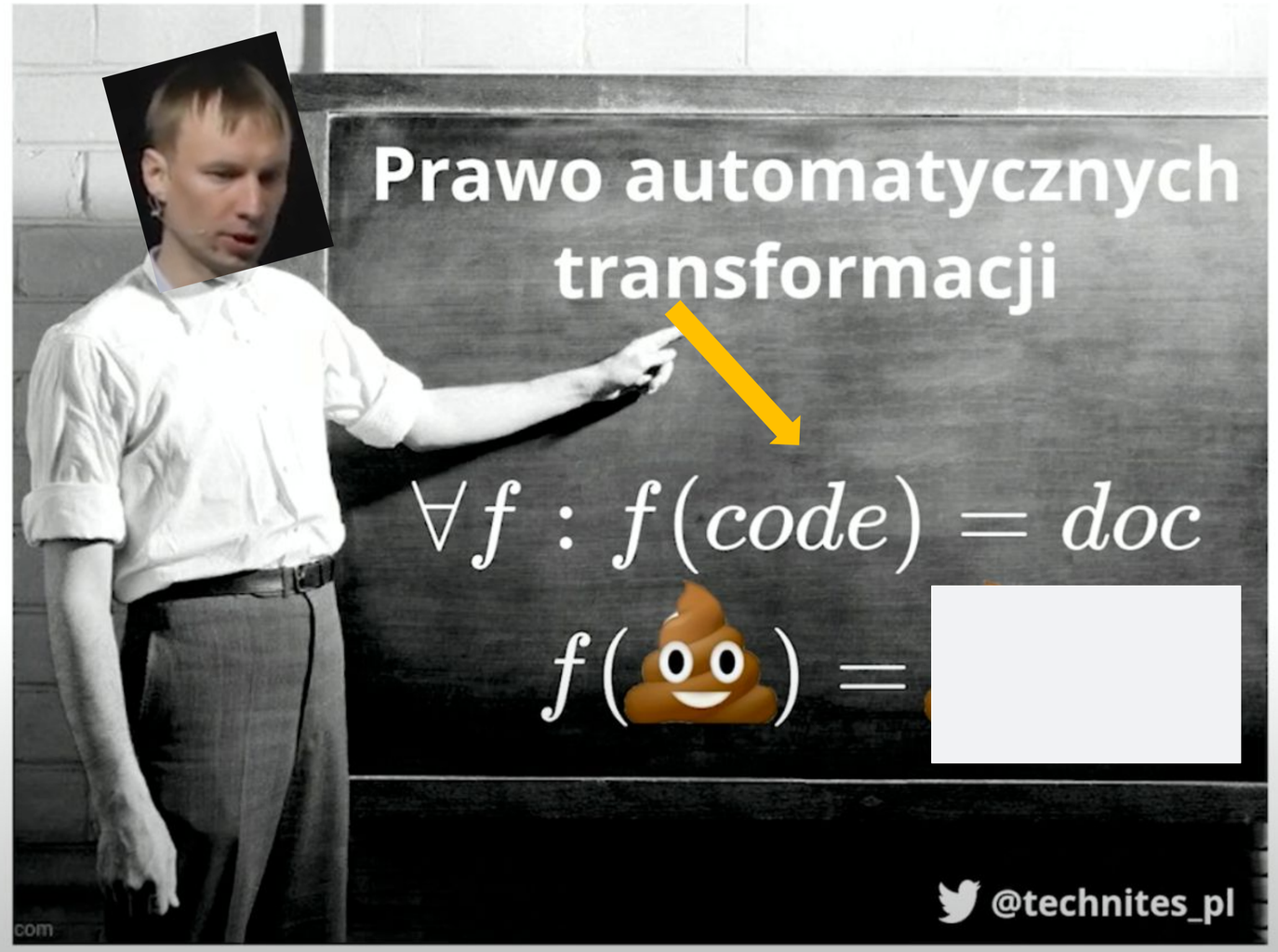


17:49 / 55:49

BoilingFrogs

DOKUMENTACJA, KTÓRA SAMA SIĘ PISZE

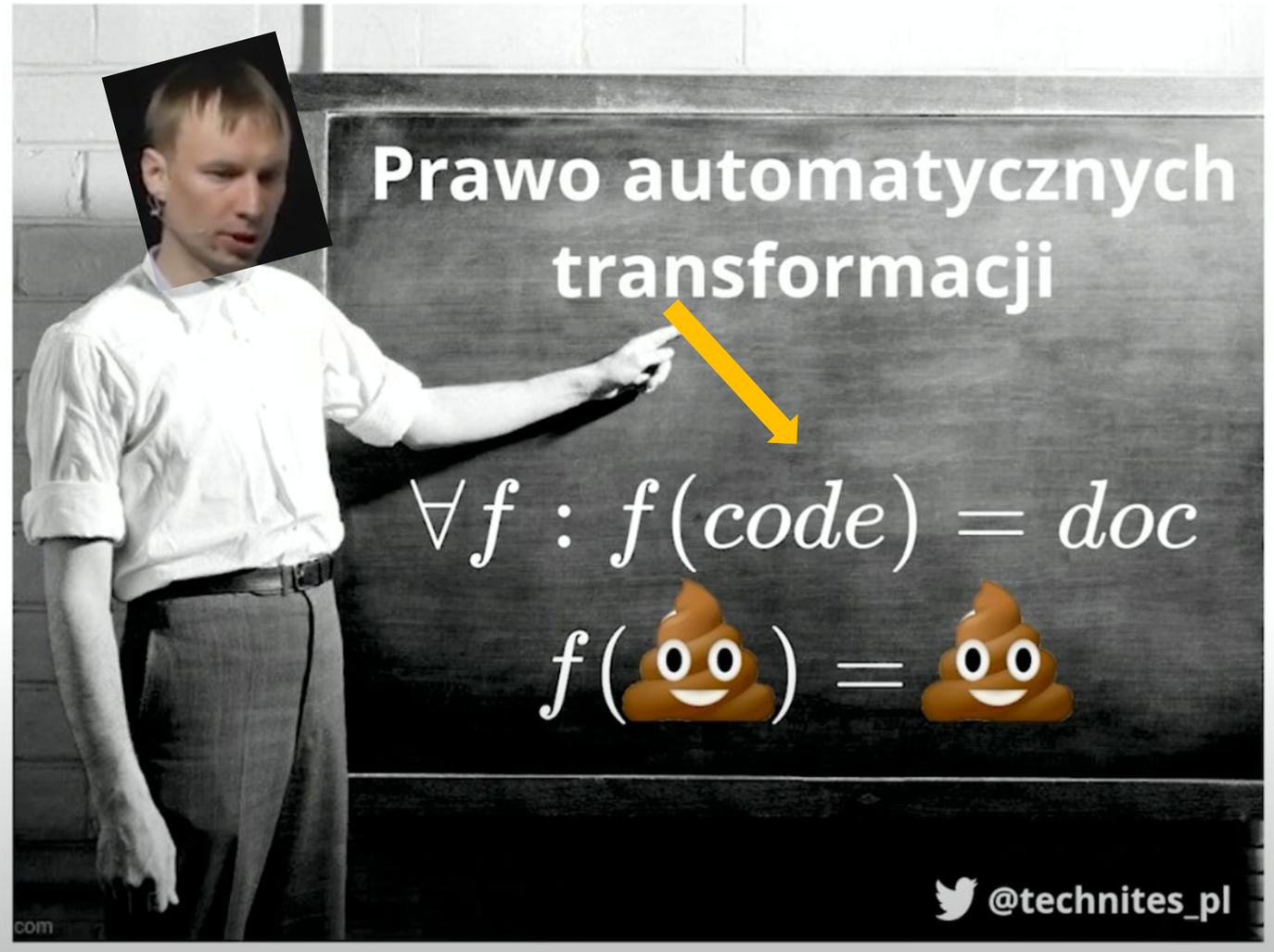
MARCIN MARKOWSKI
Software Architect and Trainer
TECHNITES_PL



BoilingFrogs

DOKUMENTACJA, KTÓRA SAMA SIĘ PISZE

MARCIN MARKOWSKI
Software Architect and Trainer
TECHNITES_PL



17:49 / 55:49

P3-model

Public

Watch 17 Fork 1 Starred 38

main 2 branches 0 tags Go to file Add file Code

technites-pl .gitignore d13105b on Apr 21 17 commits

.gitignore	.gitignore	5 months ago
Elements.md	meta model draft	5 months ago
LICENSE	Update LICENSE	6 months ago
MetaModel.md	meta model draft	5 months ago
README.md	Update README.md	5 months ago

README.md

P3 Model

License CC BY-SA 4.0

P3 Model is a tool to navigate throughout complex software systems from all important perspectives at all level of details for everyone involved in system development including nontechnical people.

About

P3 Model is a tool to navigate throughout complex software systems

Readme

CC-BY-SA-4.0 license

Activity

38 stars

17 watching

1 fork

Report repository

Releases

No releases published

Packages

No packages published

	  	   	
Domain			

	  	   	
Domain			
Technology			

	  	   	
Domain			
Technology			
People			

60. O technikach Living Documentation i modelu P3 z Marcinem Markowskim

MAY 16TH, 2023 | 01:10:20 | E60

SHARE

EMBED

RECAST

SUBSCRIBE

EPISODE SUMMARY

Istnieją trzy rodzaje dokumentacji. Przy czym pierwszy rodzaj to taki, który... nie istnieje. A o dwóch pozostałych dowiesz się z tego odcinka.

EPISODE NOTES

Istnieją trzy rodzaje dokumentacji. Przy czym pierwszy rodzaj to taki, który... nie istnieje. A o dwóch pozostałych dowiesz się z tego odcinka.

Dziś moim gościem jest Marcin Markowski, a rozmawiać będziemy o dokumentacji i sposobach na utrzymanie jej aktualności. Bo niestety, mało co tak przeszkadza podczas pracy jak dokumentacja, na której nie można polegać.

W tym odcinku rozmawiamy z Marcinem m.in. o:

- co i dlaczego warto dokumentować podczas prac nad projektem,
- typowych problemach z dokumentacją, w tym
- koncepcie Living Documentation autorstwa Cyrille Martraire,
- strategiach i konwencjach pozwalających utrzymać aktualność dokumentacji wbudowanej w projekt,
- założeniach modelu P3 i różnych perspektywach dokumentacji.

Files

master

Go to file t

- Build
- Docs/P3
  - DocsGenerator
  - JsonOutput
  - MermaidOutput
    - Domain
      - Concepts
      - Glossary
        - Domain_Glossary.md
        - Order.md
      - Processes
      - Risk management
      - Sale
        - BusinessProcesses.md
    - People
      - BusinessOrganizationalUnits
      - DevelopmentTeams
    - Technology/DeployableUnits
      - DeployableUnits.md
      - EcommerceMonolith.md
    - README.md
  - Sources
    - Contacts/Contacts
    - Payments
    - ProductsDelivery
    - RiskManagement
    - Sales
    - Startup
    - TechnicalStuff
    - DomainVisionStatement.md
  - .gitignore

ecommerce-monolith

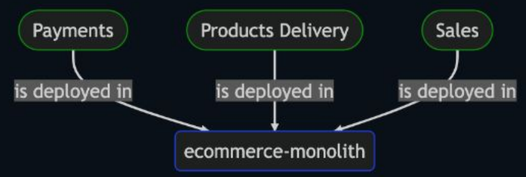
Deployable Unit

This view contains details information about ecommerce-monolith deployable unit, including:

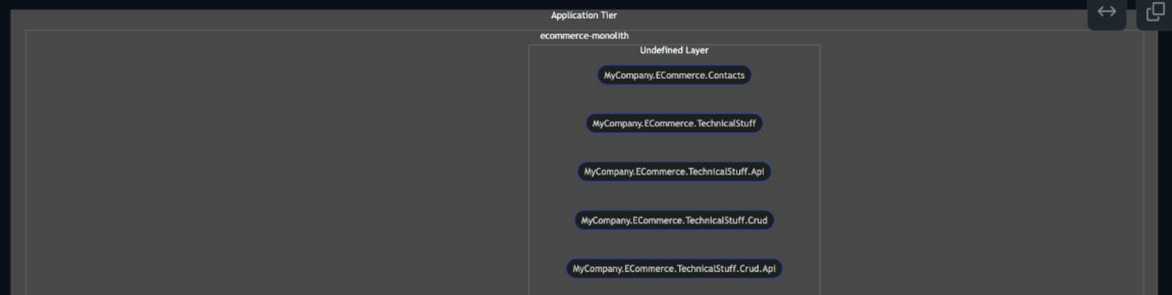
- related domain modules
- related development teams

Domain Perspective

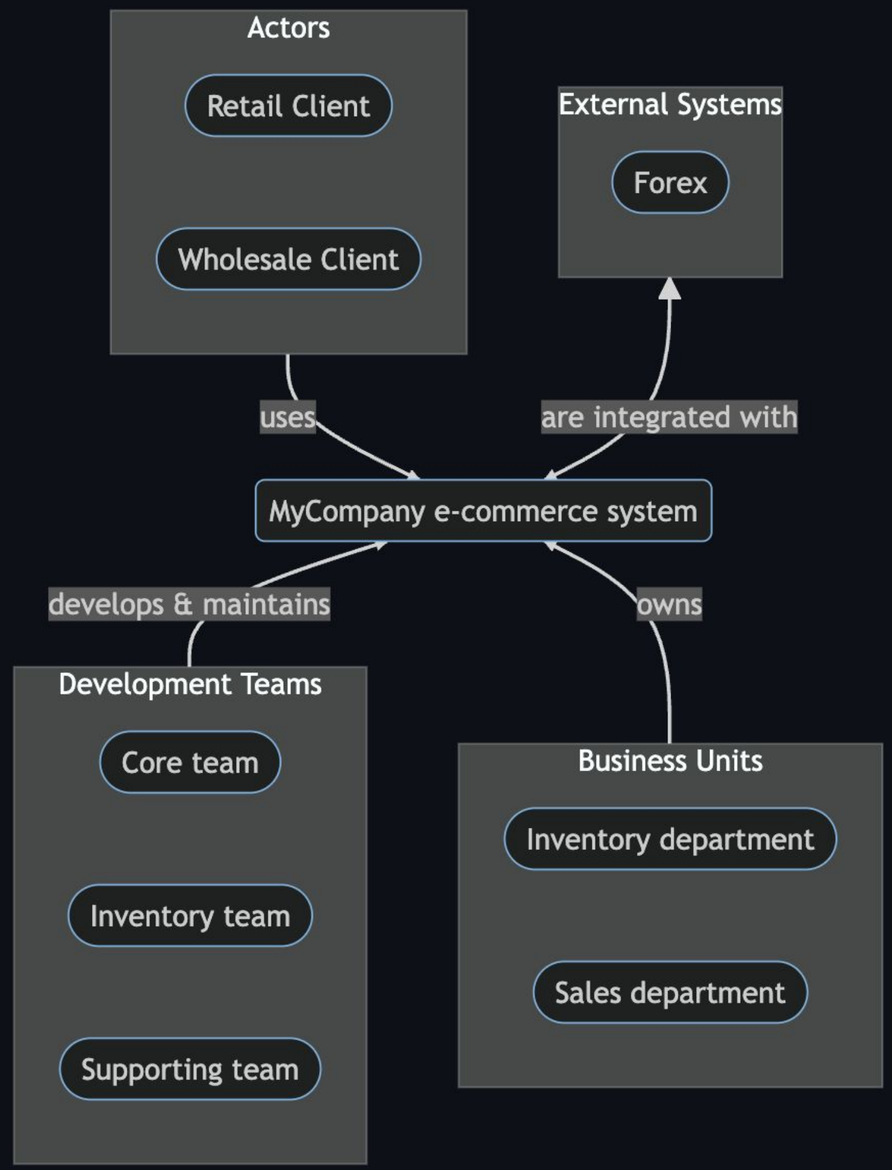
Related domain modules

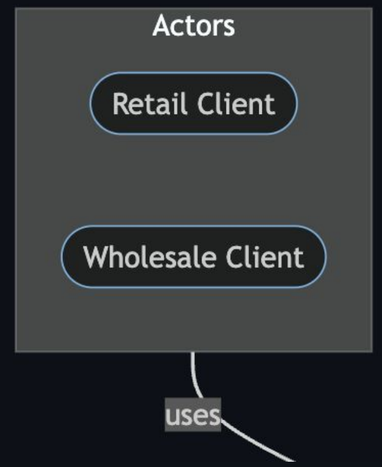


Technology Perspective



System Landscape




Actors
Retail Client
Wholesale Client
uses

uses

SEARCH

[Actor(Actors.RetailClient)] Aa ab .*

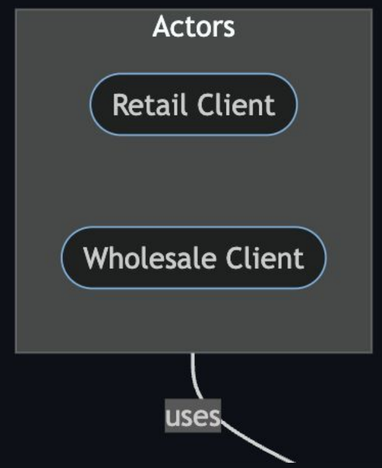
2 results in 2 files - Open in editor

PriceCartHandler.cs Sources/Sales/Sales.Proc... 1

[Actor(Actors.RetailClient)]

PlaceOrderHandler.cs Sources/Sales/Sales.Pro... 1

[Actor(Actors.RetailClient)]




```
Marcin Markowski, last month | 1 author (Marcin Markowski)
[Actor(Actors.RetailClient)]
1 reference
public class PlaceOrderHandler : CommandHandler<PlaceOrder, OrderPlaced>
{
    2 references
```

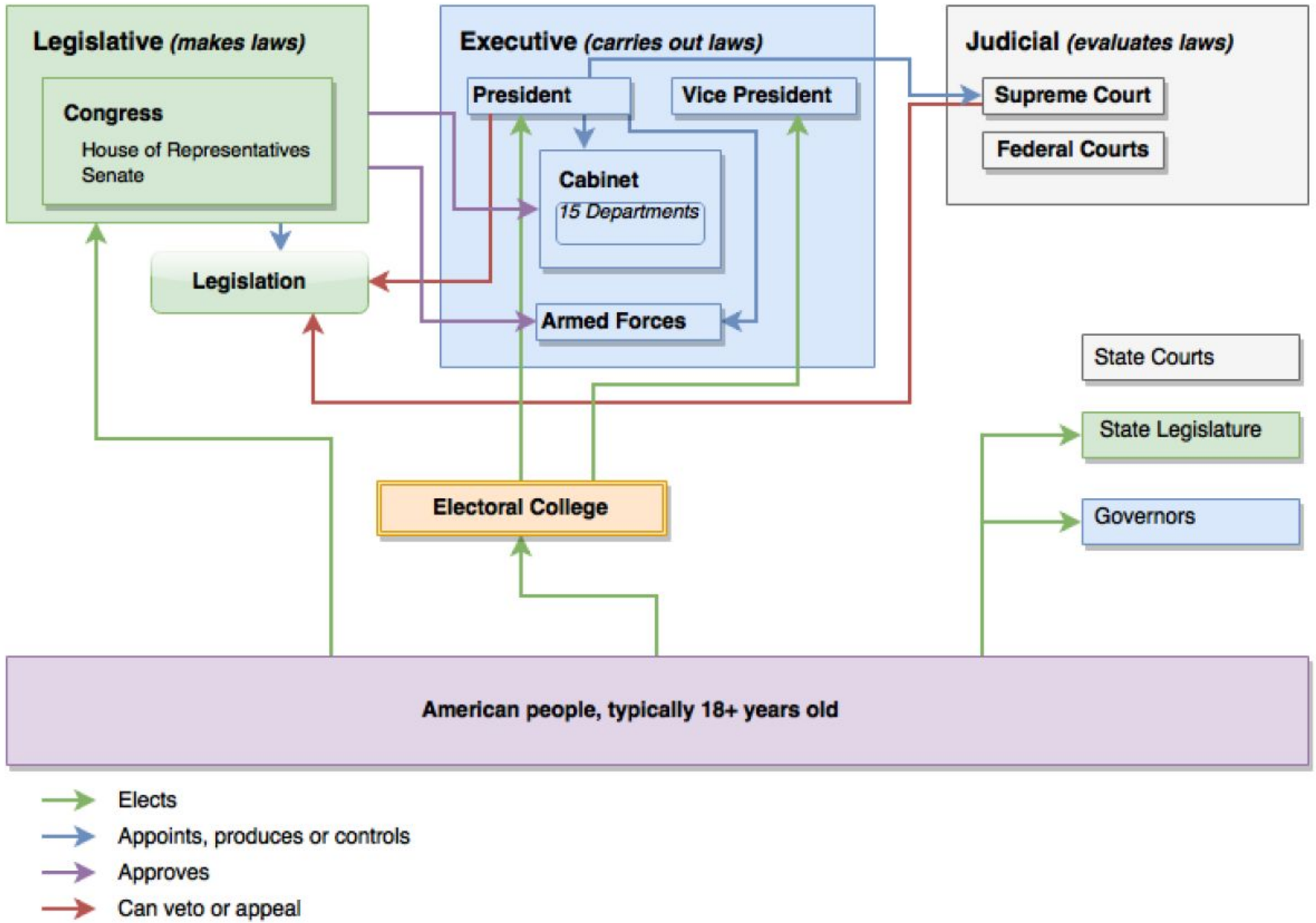
```
await P3
    .Product(product => product.UseName("MyCompany e-commerce"))
    .Repositories(repositories => repositories.Use(repositoryPath))
    .Analyzers(analyzers => analyzers
        .UseDefaults(options => options
            .TreatNamespacesAsDomainModules(namespaces => namespaces
                .OnlyFromAssembliesAnnotatedWith<DomainModelAttribute>()
                .RemoveRootNamespace("MyCompany.ECommerce"))))
    .OutputFormat(formatters => formatters
        .UseMermaid(options => options
            .Directory($"{outputPath}/MermaidOutput")
            .UseDefaultPages())
    .Analyze();
```

using text

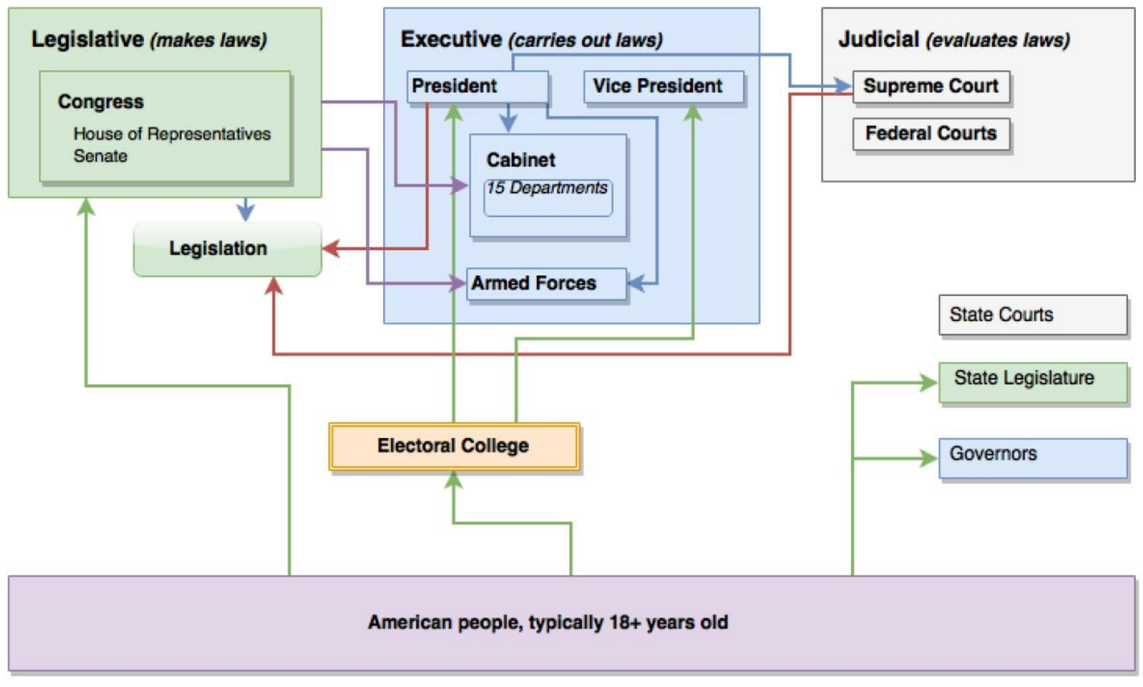
The American political system is broken into three parts. The Legislative branch consists of the Congress and the Senate and proposes legislation. Adapted from Political System of the United States, by 111Alleskönner (CC BY-SA 3.0 DE)
Legislators are elected by the American people. The Executive branch carries out the legislation, and consists of the President, the Vice President, the Cabinet and the Armed Forces. The Cabinet represents 15 departments, and these representatives are appointed directly by the President, but need to be approved by Congress. The Judicial branch evaluates legislation. The supreme court representatives are appointed directly by the President. Both the President and the Supreme Court can veto or appeal legislation. The American people elect people into the Electoral College, who then elect the President, and the Vice President. In addition to the federal political system, states have their own courts, legislators and governors.

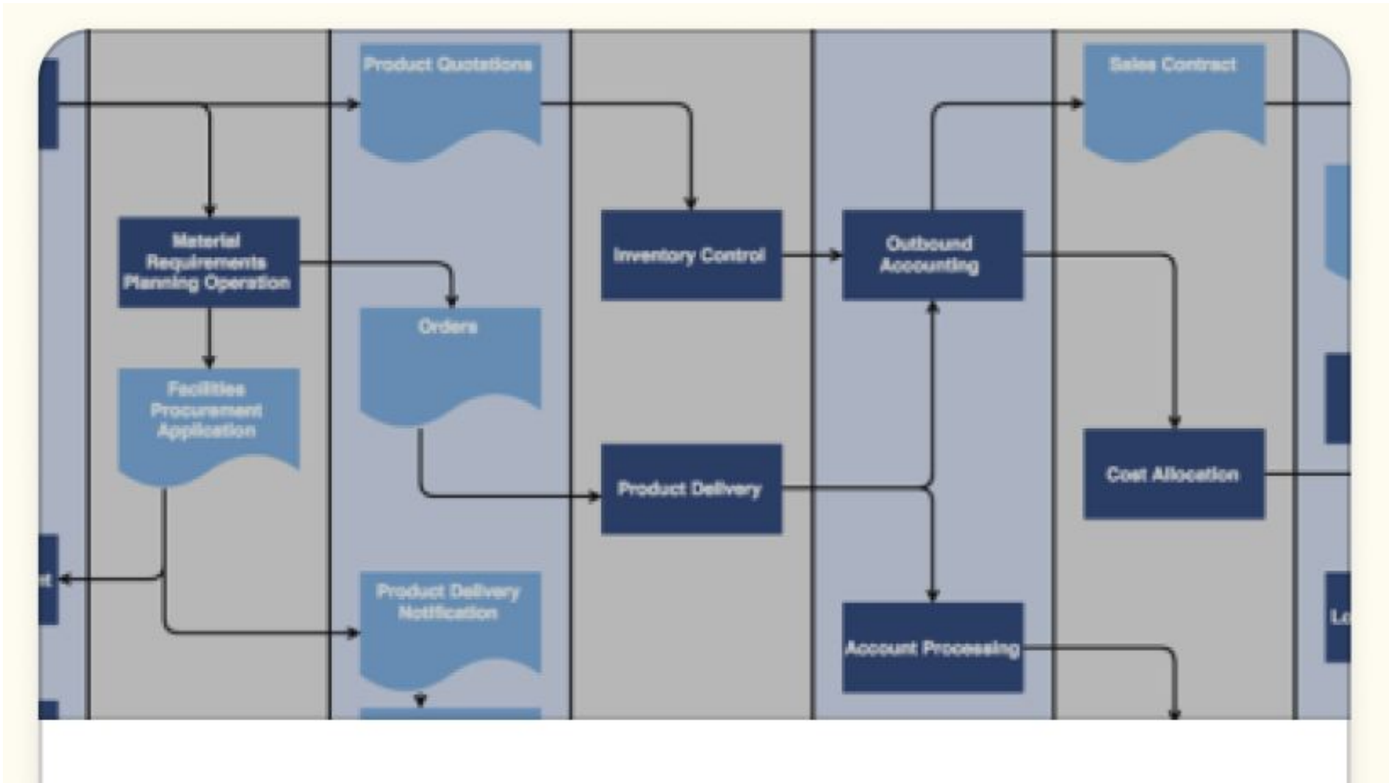
vs

American Political System



The American political system is broken into three parts. The Legislative branch consists of the Congress and the Senate and proposes legislation.
Adapted from Political System of the United States, by 111Alleskönner (CC BY-SA 3.0 DE)
Legislators are elected by the American people. The Executive branch carries out the legislation, and consists of the President, the Vice President, the Cabinet and the Armed Forces. The Cabinet represents 15 departments, and these representatives are appointed directly by the President, but need to be approved by Congress. The Judicial branch evaluates legislation. The supreme court representatives are appointed directly by the President. Both the President and the Supreme Court can veto or appeal legislation. The American people elect people into the Electoral College, who then elect the President, and the Vice President. In addition to the federal political system, states have their own courts, legislators and governors.

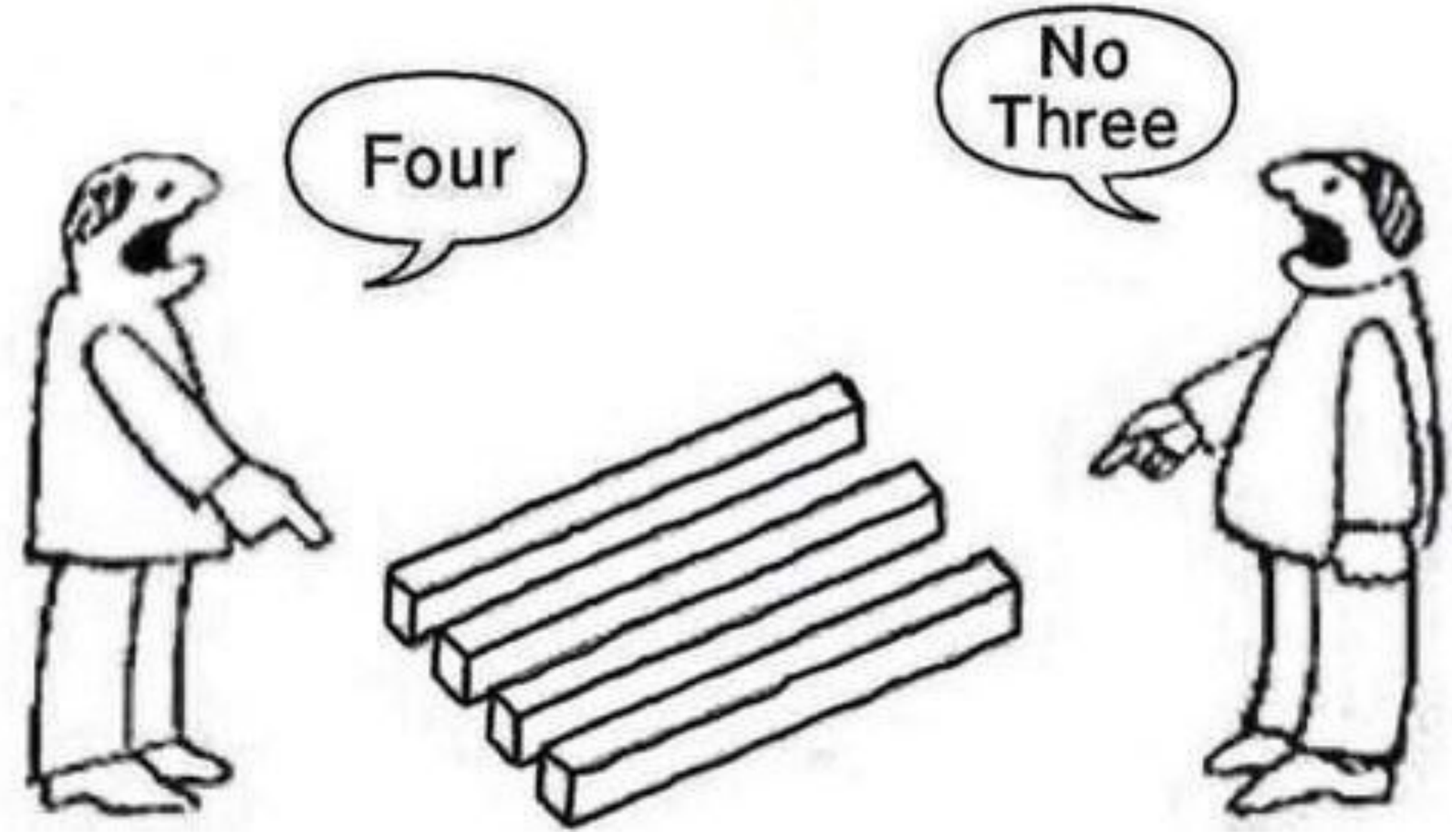


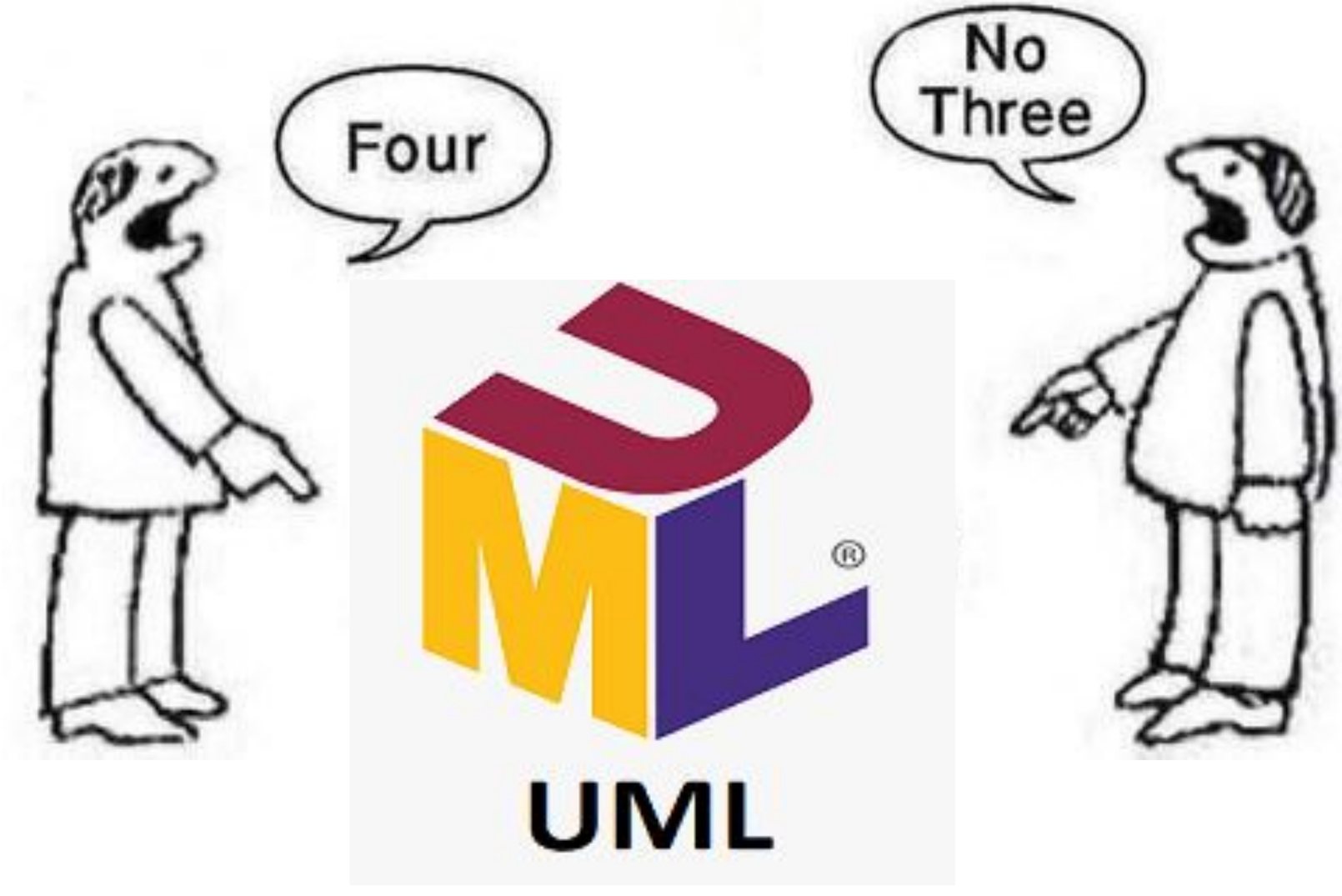


Why Are Diagrams So Powerful?

Visual communication has quickly risen to be the dominant form of communic...

drawio-app.com


Four
No
Three


Four
No
Three
UML

draw.io

draw.io

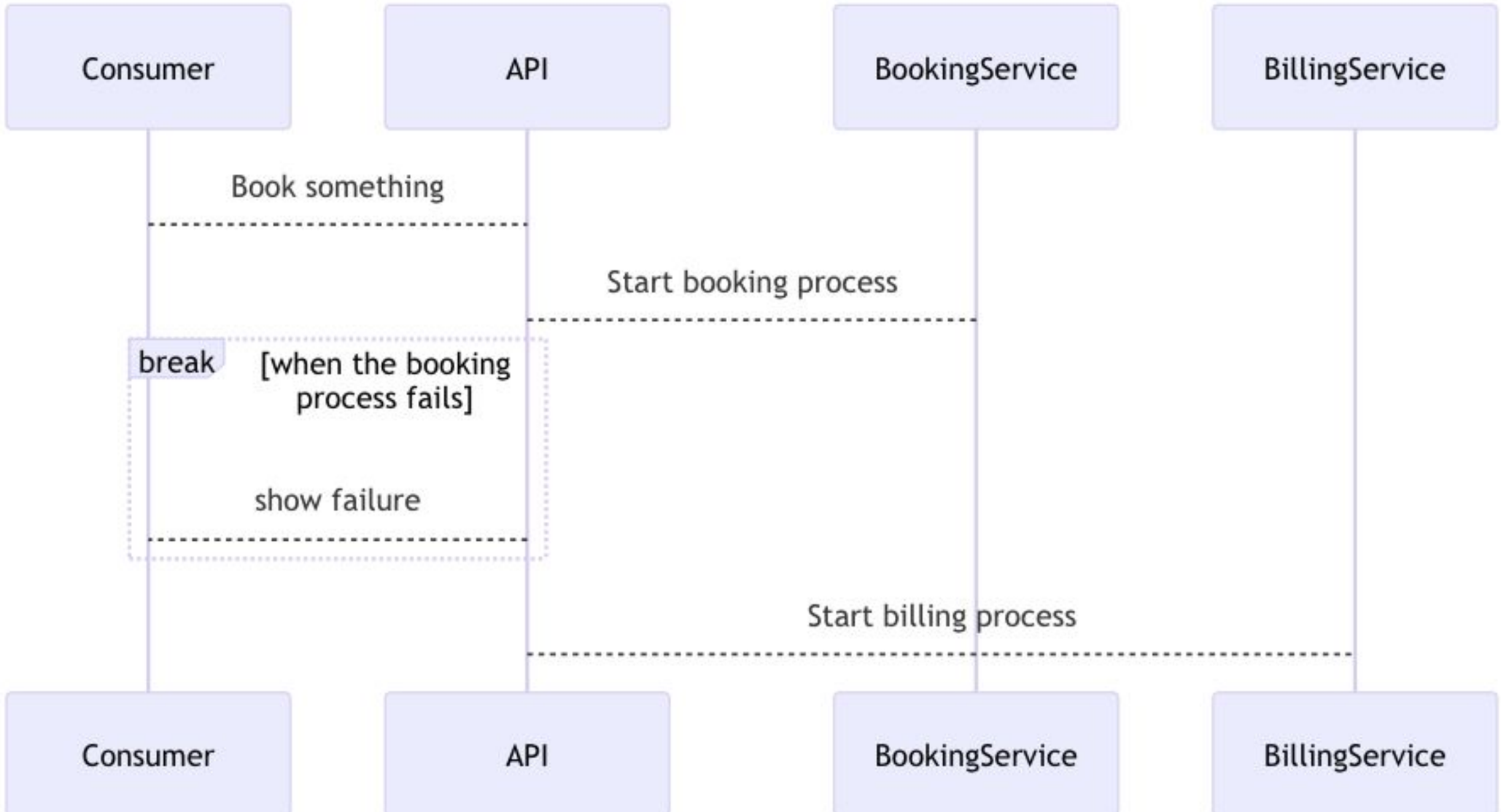
mermaid

draw.io

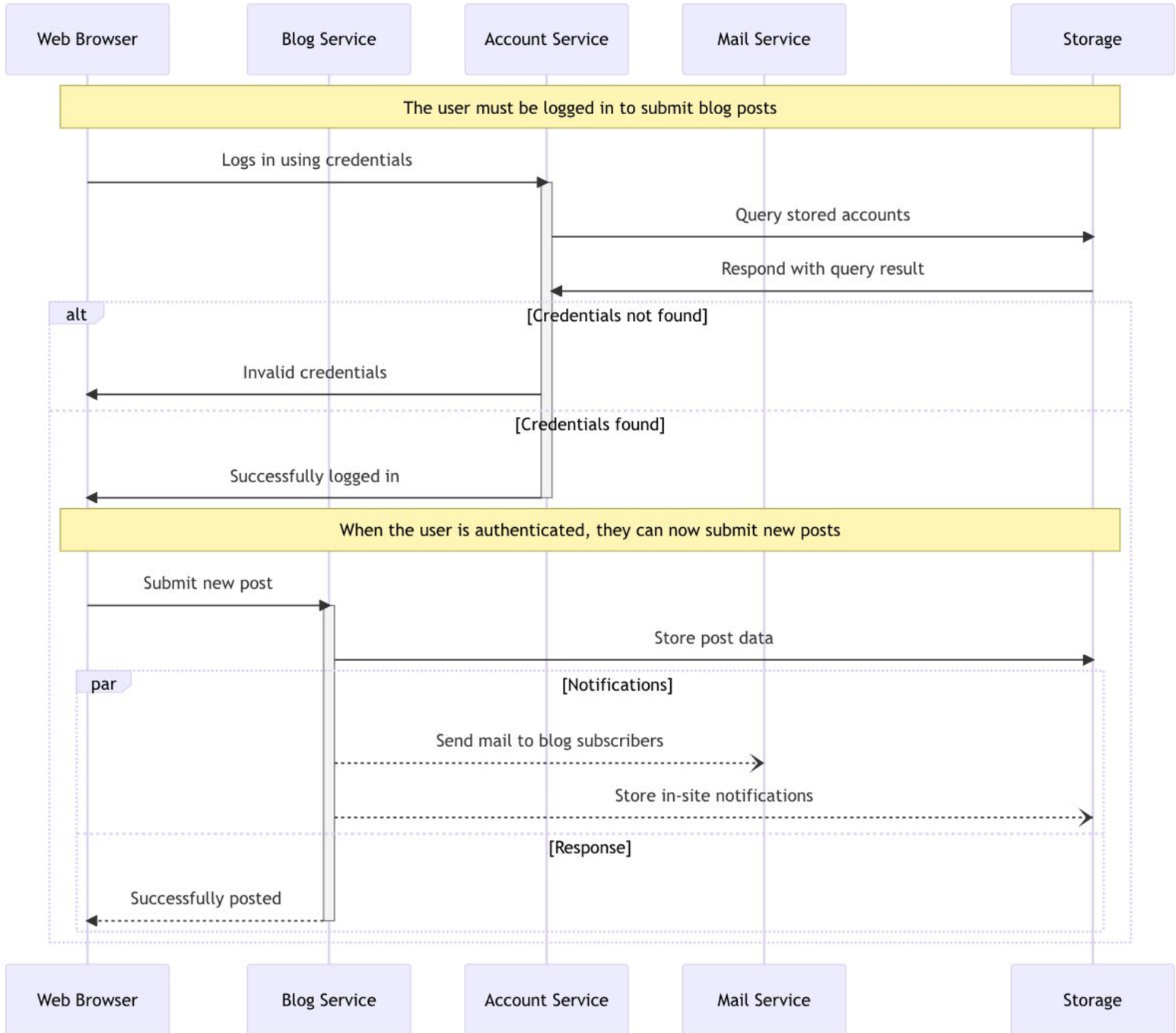
mermaid

plantuml

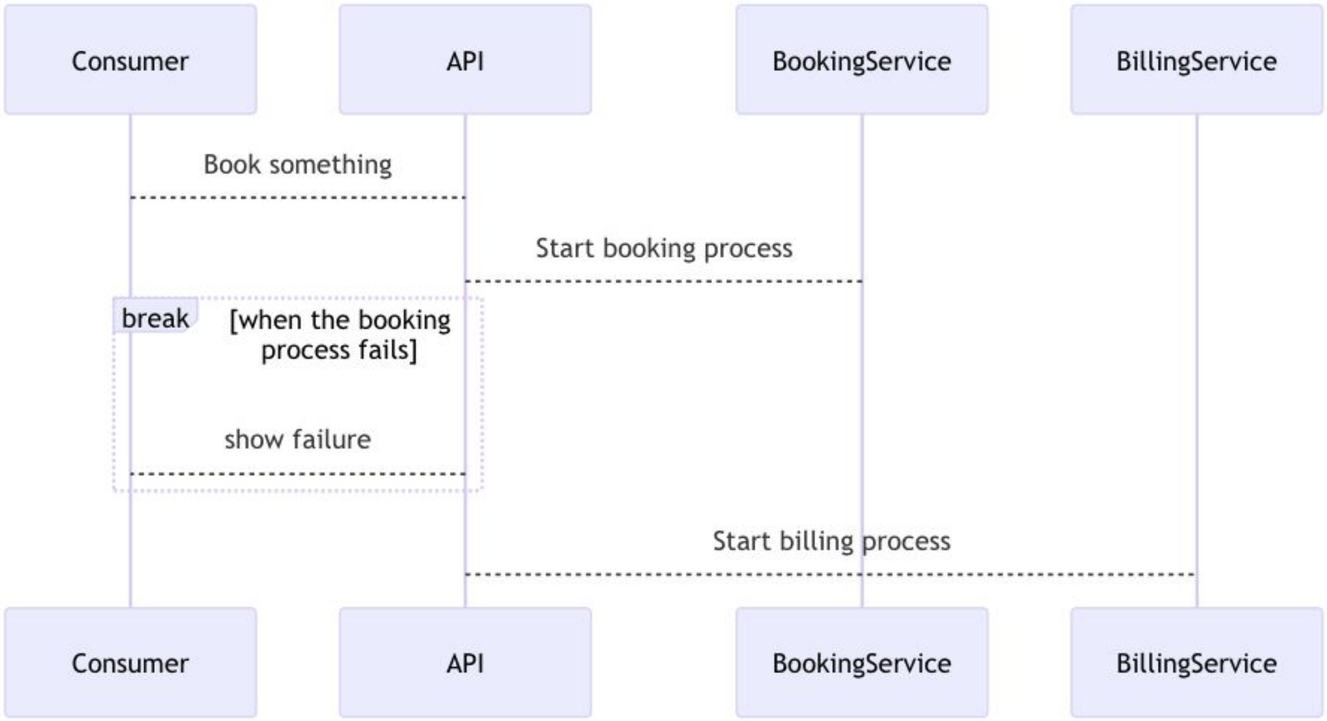
d

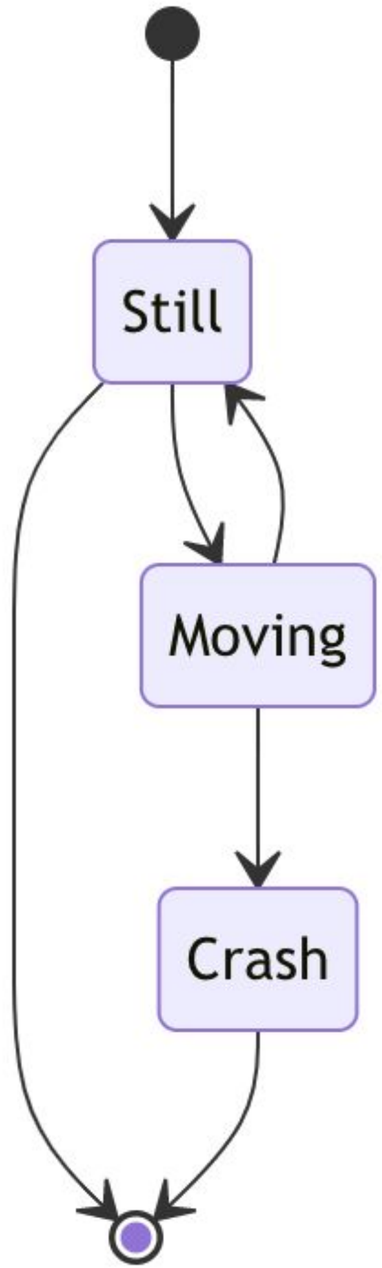

Consumer
API
BookingService
BillingService
Book something
Start booking process
break
[when the booking process fails]
show failure
Start billing process
Consumer
API
BookingService
BillingService

```mermaid
sequenceDiagram
    Consumer-->>API: Book something
    API-->>BookingService: Start booking process
    break when the booking process fails
        API-->>Consumer: show failure
    end
    API-->>BillingService: Start billing process
```


Web Browser
Blog Service
Account Service
Mail Service
Storage
The user must be logged in to submit blog posts
Logs in using credentials
Query stored accounts
Respond with query result
alt
[Credentials not found]
Invalid credentials
[Credentials found]
Successfully logged in
When the user is authenticated, they can now submit new posts
Submit new post
Store post data
par
[Notifications]
Send mail to blog subscribers
Store in-site notifications
[Response]
Successfully posted
Web Browser
Blog Service
Account Service
Mail Service
Storage

```mermaid
sequenceDiagram
    Consumer-->API: Book something
    API-->BookingService: Start booking process
    break when the booking process fails
        API-->Consumer: show failure
    end
    API-->BillingService: Start billing process
```




Still
Moving
Crash

```mermaid
stateDiagram
    [*] --> Still
    Still --> [*]

    Still --> Moving
    Moving --> Still
    Moving --> Crash
    Crash --> [*]
```

High-level overview

Principal

High-level overview

Principal

Me

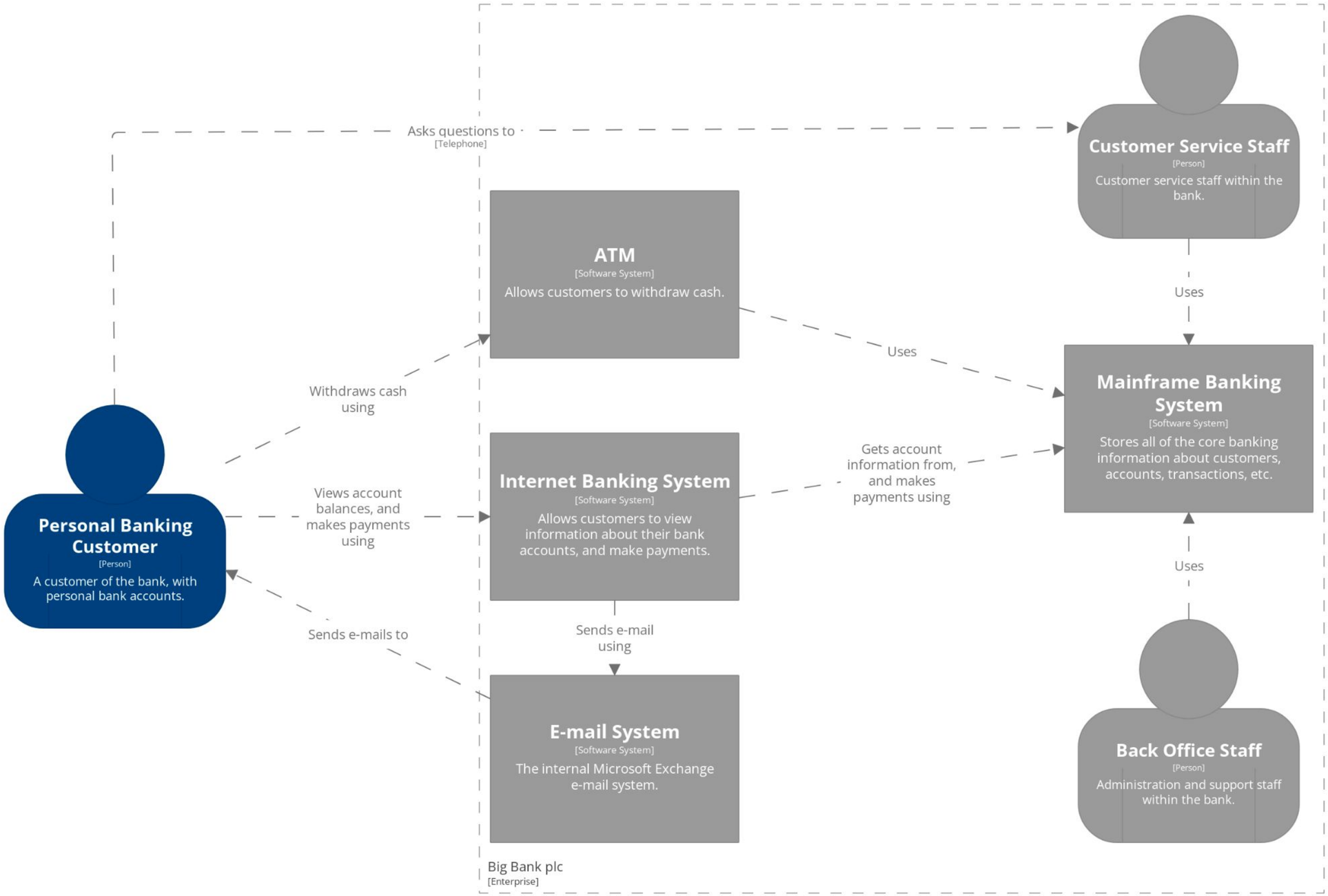
High-level overview

Principal

Me

High-level overview

New hire


Asks questions to
[Telephone]
Customer Service Staff
[Person]
Customer service staff within the bank.
ATM
[Software System]
Allows customers to withdraw cash.
Uses
Withdraws cash using
Uses
Mainframe Banking System
[Software System]
Stores all of the core banking information about customers, accounts, transactions, etc.
Personal Banking Customer
[Person]
A customer of the bank, with personal bank accounts.
Views account balances, and makes payments using
Internet Banking System
[Software System]
Allows customers to view information about their bank accounts, and make payments.
Gets account information from, and makes payments using
Uses
Sends e-mails to
Sends e-mail using
E-mail System
[Software System]
The internal Microsoft Exchange e-mail system.
Back Office Staff
[Person]
Administration and support staff within the bank.
Big Bank plc
[Enterprise]

Architecture

Decision Records

Architecture

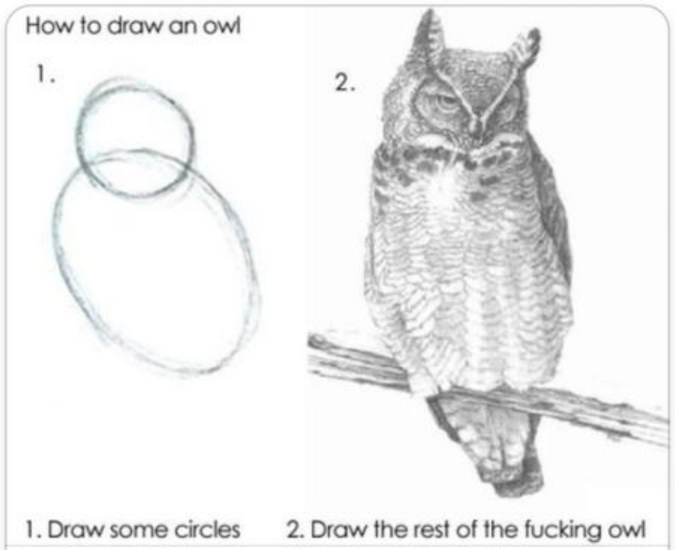
Decision Records

no 'bad' decisions; just unrecorded

Architecture

Decision Records

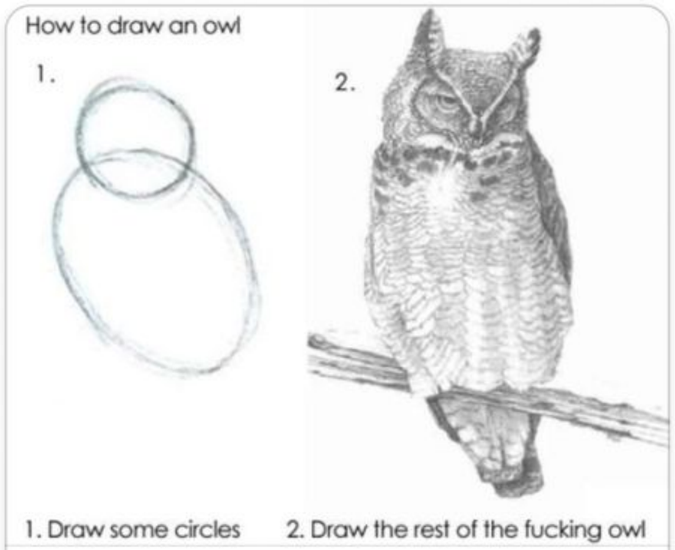
no 'bad' decisions; **just unrecorded**



Design Docs at Google

One of the key elements of Google's software engineering culture is the...

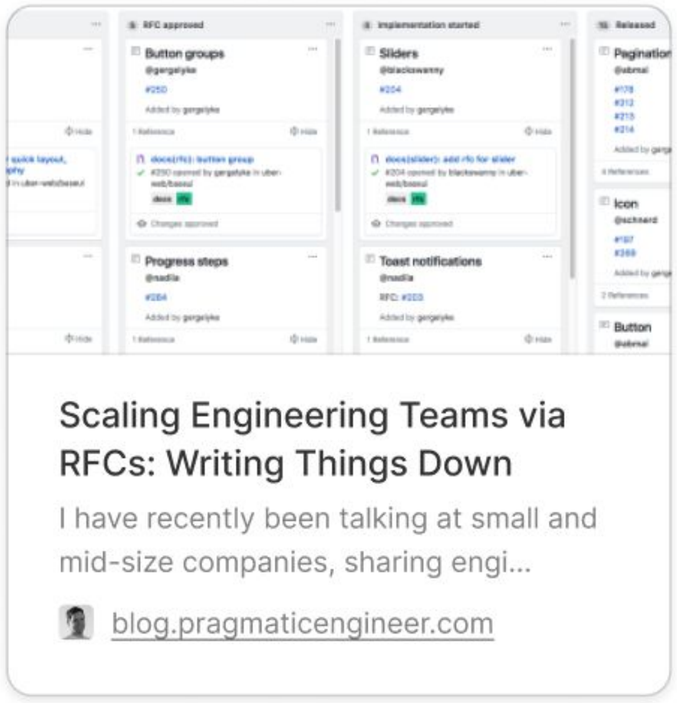
industrialempathy.com



Design Docs at Google

One of the key elements of Google's software engineering culture is the...

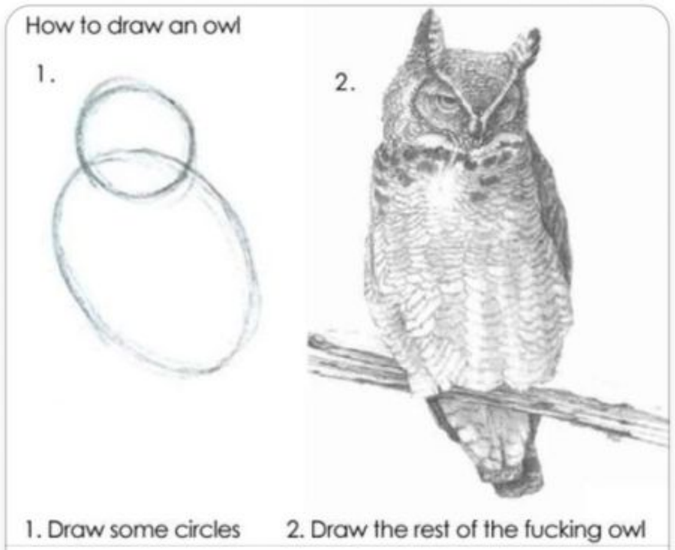
industrialempathy.com



Scaling Engineering Teams via RFCs: Writing Things Down

I have recently been talking at small and mid-size companies, sharing engi...

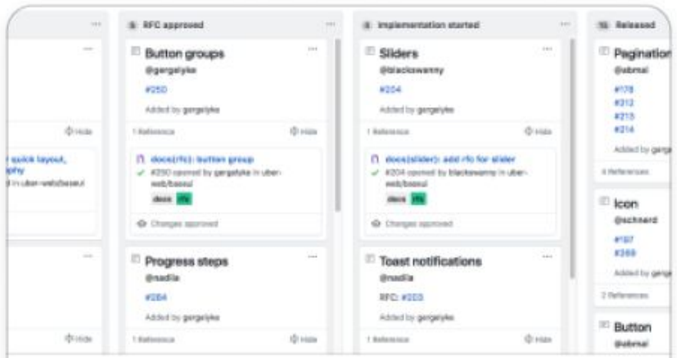
blog.pragmaticengineer.com



Design Docs at Google

One of the key elements of Google's software engineering culture is the...

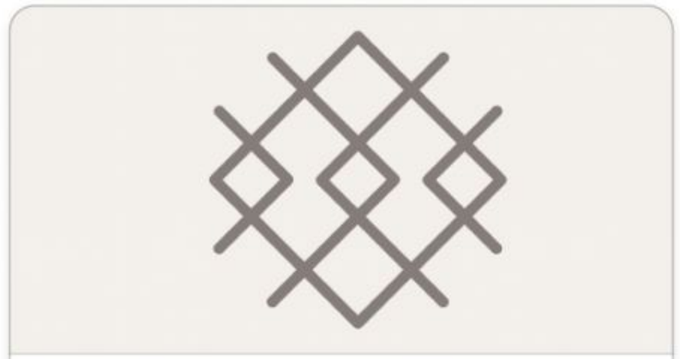
industrialempathy.com



Scaling Engineering Teams via RFCs: Writing Things Down

I have recently been talking at small and mid-size companies, sharing engi...

blog.pragmaticengineer.com



Technical Decision-Making and Alignment in a Remote Culture |...

In a world of asynchronous communication, it's more important t...

multithreaded.stitchfix.com

1.

1.Context and Problem

1.Context and Problem

2.What We Want (and the Deciding Factors)

1. Context and Problem
2. What We Want (and the Deciding Factors)
3. What Options We Have (and Their Consequences)

1. Context and Problem
2. What We Want (and the Deciding Factors)
3. What Options We Have (and Their Consequences)
4. Decision

Architecture Decision Record: PostgreSQL database

Title: Use of PostgreSQL Database

Context

Our organization is currently evaluating options for the most suitable database management system to be used in our upcoming project. We have narrowed our search down to two options – PostgreSQL and MySQL. After careful consideration, we have decided to use PostgreSQL due to its advanced features and performance capabilities.

Decision

We will use PostgreSQL as the preferred database management system for our organization's upcoming project.

Justification

After analyzing and comparing the features of both PostgreSQL and MySQL, we have come to the conclusion that PostgreSQL is the better option because of the following reasons:

1. **Advanced features:** PostgreSQL has a wide range of advanced features like JSON support, full-text search, and spatial data management, which are essential for our project.
2. **Performance:** PostgreSQL has a proven track record of excellent performance capabilities, which will ensure that our project runs smoothly without any performance issues.
3. **Community support:** PostgreSQL has a large and active community that provides regular updates and support, making it easier to resolve any issues that arise during the project development phase.
4. **Scalability:** PostgreSQL is highly scalable, and it can handle large amounts of data with ease, which will be essential for our project.
5. **Security:** PostgreSQL has a robust security mechanism that will ensure that our project data is secure at all times, and it meets all our organization's security requirements.

Alternatives

The other option that we considered was MySQL. MySQL is also a highly capable database management system, but it lacks some of the advanced features that PostgreSQL offers, and its performance capacity is not as good as that of PostgreSQL.

Conclusion

Based on our analysis, we have decided to use PostgreSQL as the preferred database management system for our upcoming project. This decision has been made after careful consideration, and we believe that it will help us achieve our project goals efficiently and effectively.

Credit: this page is generated by ChatGPT, then edited for clarity and format.

ADR-0044: Duty devs responsibilities

Status: accepted
Deciders: Lukasz Dziekan, Tomasz Szczerba, Mariusz Budzyn, Jakub Dropia, Marcin Malinowski, Marcin Liszewski, Mateusz Pobudejski, Paweł Marcinkowski, Piotr Wiśniewski Date: 2021-04-14
Technical Story: Currently devs responsibilities are unclear

Context and Problem Statement

We have a few following problems:

- it is unclear what developer should or should not do with incidents
- it is unclear whether duty dev should acknowledge and work on incident, if another already acknowledged it
- it is unclear **when** an incident should be **resolved**

Decision Drivers

- we need to maintain support level (we do not cut scope)
- procedures should be clear and as simple as possible
- we need to know whether the solution is good or bad

Current situation

- developers handle all PD calls
- Each monitor always triggers wakeup call
- duty devs more often than not handle #help-allegro-pay issues
- some devs analize and/or fix code affecting few users at night
- some devs almost do not react at night, if the issue is not critical
- current metrics situation

Current understanding of duty devs responsibilites:

- official instructions are here
- not everyone handles tickets in the same manner
- there is no official information about WHO handles **#help-allegro-pay**
- there is no procedure how to create proper bug if duty dev can't fix the issue with provided tools mentioned in instructions, leading to handovers of vaguely described issues with close to none analysis.
- there is no instruction when an incident can be resolved

Considered Improvements

1. Write down **flow for handling PD alerts**.
2. We should **improve monitoring on lower environments** to detect potential issues faster
3. Propose **new, slightly improved responsibilities**

Considered Improvements

1. Write down **flow for handling PD alerts**.
2. We should **improve monitoring on lower environments** to detect potential issues faster
3. Propose **new, slightly improved responsibilities**

Conclusion and suggested approach:

- Babajaga takes care of #help-allegro-pay
- Take duty manual and incident workflow ***seriously***
- Fix duty on UAT
  - fix monitoring to not call PD, but create slack alerts
  - duty dev is not expected to fix issues on UAT, but monitor if all issues are taken care of - it can be checked once or twice a day if all issues are taken care of on slack by having proper icon added.
- Remove duty on DEV
- Implement the above for 2 months, and check metrics again.

Expected improvements in metrics:

- HAP
  - clarity of duty responsibilities
  - understanding duty responsibilities
  - are duty responsibilities too narrow/too wide
  - organization of duty
- HLPA
  - reduced time to acknowledge for p90
- HLPR
  - reduced time to resolve for p90

What needs to be done:

- Babajaga takes course in bookkeeping 101 - align with Firewatch
- Prepare workflow on #allegropay-allerts-prod to hand over duty and keep track of overtime hours in excel sheet **without keeping track who, when and how many** overtime hours had
- Announce that Babajaga takes care of #help-allegro-pay
- Announce new duty responsibilities
- Announce incident workflow manual along with PD Inc Work Item Type and incident dashboard
- Change monitoring on UAT to generate slack notifications
- Remove monitoring on DEV

.NET 7 in Allegro Pay

.NET 7 release date is November 2022. The LTS version remains .NET 6, and the next one will be .NET 8 next year. Therefore, in order not to unnecessarily use developers' time, the Aard team has decided not to carry out a global migration of services to .NET 7.

Rules regarding .NET 7:

- We do not ban the use of .NET 7, but you do so at your own risk - Aard's support for this will be limited.
- Your migration cannot force migration to other teams (if you are issuing nuget packages for other services / teams, they must target .NET 6).
- By targeting .NET 6 in each case we mean both runtime and core Microsoft.* And System.* packages (they follow the same publishing cycle as the whole .NET, and a higher major version may introduce breaking changes). List of packages is available here.
- To reduce the impact of a possible mistake, before the .NET 7 release, there is an additional build step, which will check if the .NET 6 targeting package does not contain core packages in the 7.x version (but while targeting .NET 7, we can use 7.x packages). For recommendations on pinning the package versions, see recommendations below.
- The package versions in group Build of `paket.lock` are irrelevant - 7.y.z is acceptable here.

In case of any questions or doubts, we remain at your disposal at #help-aard.

Package version pinning - recommendations

To prevent bumping MS packages to 7.0.0 consider following recommendations:

1. Use `paket update --keep-major` flag to prevent bumping from 6.y.z to 7.0.0. Note: this will pin all packages' major versions.
2. Add following package to `paket.dependencies` (group Main):

   ```
   nuget AllegroPay.PackageVersionLock
   ```

   This will prevent paket from updating MS packages to 7.0.0. Do not add this package to any `paket.references` file, `paket.dependencies` is enough.

ADR != RFC

ADR <> RFC

More use cases

More use cases

Playbooki / FAQ - on call

More use cases

Playbooki / FAQ - on call
Getting started

More use cases

Playbooki / FAQ - on call
Getting started
How to build, run, run tests

More use cases

Playbooki / FAQ - on call
Getting started
How to build, run, run tests
Access Management

More use cases

Playbooki / FAQ - on call
Getting started
How to build, run, run tests
Access Management
Service's clients

More use cases

Playbooki / FAQ - on call
Getting started
How to build, run, run tests
Access Management
Service's clients
Researches/PoC results

More use cases

Playbooki / FAQ - on call
Getting started
How to build, run, run tests
Access Management
Service's clients
Researches/PoC results
Summaries of tech initiatives

Tooling - “how”


When you start coding in a new language without reading the documentation

Where?

vs

Confluence

vs

MkDocs

Project documentation with Markdown.

vs

MkDocs

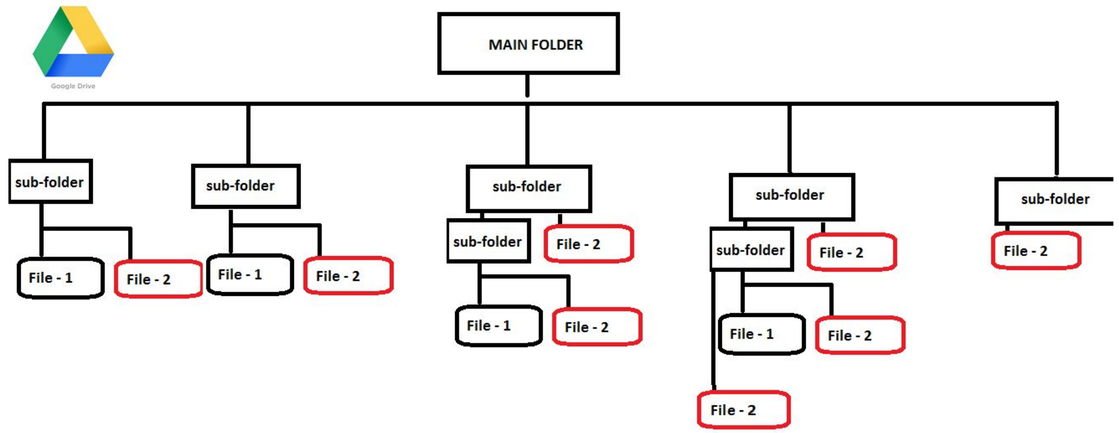
Project documentation with Markdown.

vs

VS

MkDocs

Project documentation with Markdown.



Catalog

An open platform for building developer portals

Powered by a centralized software catalog, Backstage restores order to your infrastructure and enables your product teams to ship high-quality code quickly — without compromising autonomy.

GITHUB OFFICE HOURS

All components (232)

Backstage

Search

Home

Catalog

APIs

Docs

Announcements

Pager Duty

Tool Box

Allegro Pay Catalog

CREATE

SUPPORT

Kind

Component

Type

service

PERSONAL

Owned 0

Starred 0

ALLEGRO PAY

All 20

OWNER

GROUP:CAREBEARS

LIFECYCLE

PRODUCTION

PROCESSING STATUS

All service components (20)

Filter

NAME	SYSTEM	OWNER	LIFECYCLE	DESCRIPTION	TAGS	ACTIONS
AccountService		carebears	production	Handles a range…		
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			

Backstage

Search

Home

Catalog

APIs

Docs

Announcements

Pager Duty

Tool Box

Allegro Pay Catalog

CREATE

SUPPORT

Kind

Component

Type

service

PERSONAL

Owned 0

Starred 0

ALLEGRO PAY

All 20

OWNER

GROUP:CAREBEARS

LIFECYCLE

PRODUCTION

PROCESSING STATUS

All service components (20)

Filter

NAM	SYSTEM	OWNER	LIFECYCLE	DESCRIPTION	TAGS	ACTIONS
Acc ntService		carebears	production	Handles a range…		
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			

Backstage

Search

Home

Catalog

APIs

Docs

Announcements

Pager Duty

Tool Box

Allegro Pay Catalog

CREATE

SUPPORT

Kind

Component

Type

service

PERSONAL

Owned 0

Starred 0

ALLEGRO PAY

All 20

OWNER

GROUP:CAREBEARS

LIFECYCLE

PRODUCTION

PROCESSING STATUS

All service components (20)

Filter

NAME	SYSTEM	OWNER	LIFECYCLE	DESCRIPTION	TAGS	ACTIONS
AccountService		carebears	production	Handles a range…		
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			
		carebears	production			

COMPONENT — SERVICE

AccountService ☆

Owner
carebears

Lifecycle
production

OVERVIEW | CI/CD | API | KUBERNETES | ARGO CD | DEPENDENCIES | DOCS | HEALTH CHECKS | PULL REQUESTS | GIT TAGS | AZURE | Q&A

About

VIEW SOURCE | VIEW TECHDOCS

DESCRIPTION

Handles a range of account-related tasks

OWNER
carebears

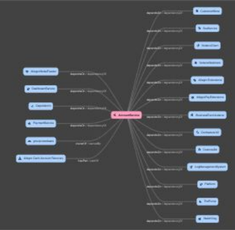
SYSTEM

TYPE
service

LIFECYCLE
production

No Tags

Relations



View graph →

Links

AccountService Dev | AccountService Uat | AccountService Xyz

Has subcomponents

NAME	OWNER	TYPE	LIFECYCLE	DESCRIPTION

No subcomponent is part of this component
Learn how to change this.

General entity health checks

Collection of general quality checks, related to standardized way to describe any component

Group Owner Check
Verifies that a Group has been set as the owner for this entity

Title Check
Verifies that a Title, used to improve readability, has been set for this entity

Entity up to date Check
Verifies that entity was confirmed up to date max 6 months ago

TechDocs Check
Verifies that TechDocs has been enabled for this entity

PagerDuty

SERVICE DIRECTORY | CREATE INCIDENT | ESCALATION POLICY

Incidents | Change Events

Nice! No incidents found!

ON CALL

Rafal Schmidt
rafal.schmidt@allegro.com

Readme

TLDR: AccountService is microservice responsible for managing account details, transaction history, generating bank statements, transferring money, handling Vodeno webhooks regarding transfers, and adding initial account details after customer onboarding.

AccountService is microservice is designed to handle various banking operations and account management tasks. Its main responsibilities include:

- Account Details: The microservice stores and manages important account information such as IBAN (International Bank Account Number), status, balance, and available funds.
- Transaction History: It maintains a record of past transactions, including details such as dates, types, groups, and categories. This allows for easy tracking and analysis of a customer's financial activities.
- Generate Statement: The microservice integrates with Vodeno, a third-party banking platform, to generate monthly bank statements. Using the provided information, it creates a statement summarizing the account's activities for a specific month.
- Transfer Money/Create Payment: After verifying the transaction with P24 (likely a payment gateway or provider), this microservice facilitates the transfer of funds or creation of payments through integration with Vodeno. This allows customers to initiate transfers securely and efficiently.
- Vodeno Webhook Integration: The microservice handles Vodeno webhooks, which are HTTP callbacks triggered by events in the Vodeno platform. Specifically, it captures webhook notifications related to changes in account balances and performs the necessary actions to update the account balance accordingly.
- Add Initial Account Details: After a customer completes the onboarding process, this microservice is responsible for adding the necessary details of the newly onboarded customer's main account. These details are essential for establishing a functional and personalized banking relationship with the customer.

Getting Started

Readme →

COMPONENT — SERVICE

AccountService ☆

Owner
carebears

Lifecycle
production

OVERVIEW | CI/CD | API | KUBERNETES | ARGO CD | DEPENDENCIES | DOCS | HEALTH CHECKS | PULL REQUESTS | GIT TAGS | AZURE | Q&A

About

VIEW SOURCE

VIEW TECHDOCS

DESCRIPTION

Handles a range of account-related tasks

OWNER

carebears

TYPE

service

LIFECYCLE

production

TAGS

No Tags

Catalog Graph

component:acash-accountservice

FILTERS

SUPPORT

MAX DEPTH

1

KINDS

API Component Domain Group +3

RELATIONS

ownerOf ownedBy consumesApi apiConsumedBy +10

Direction

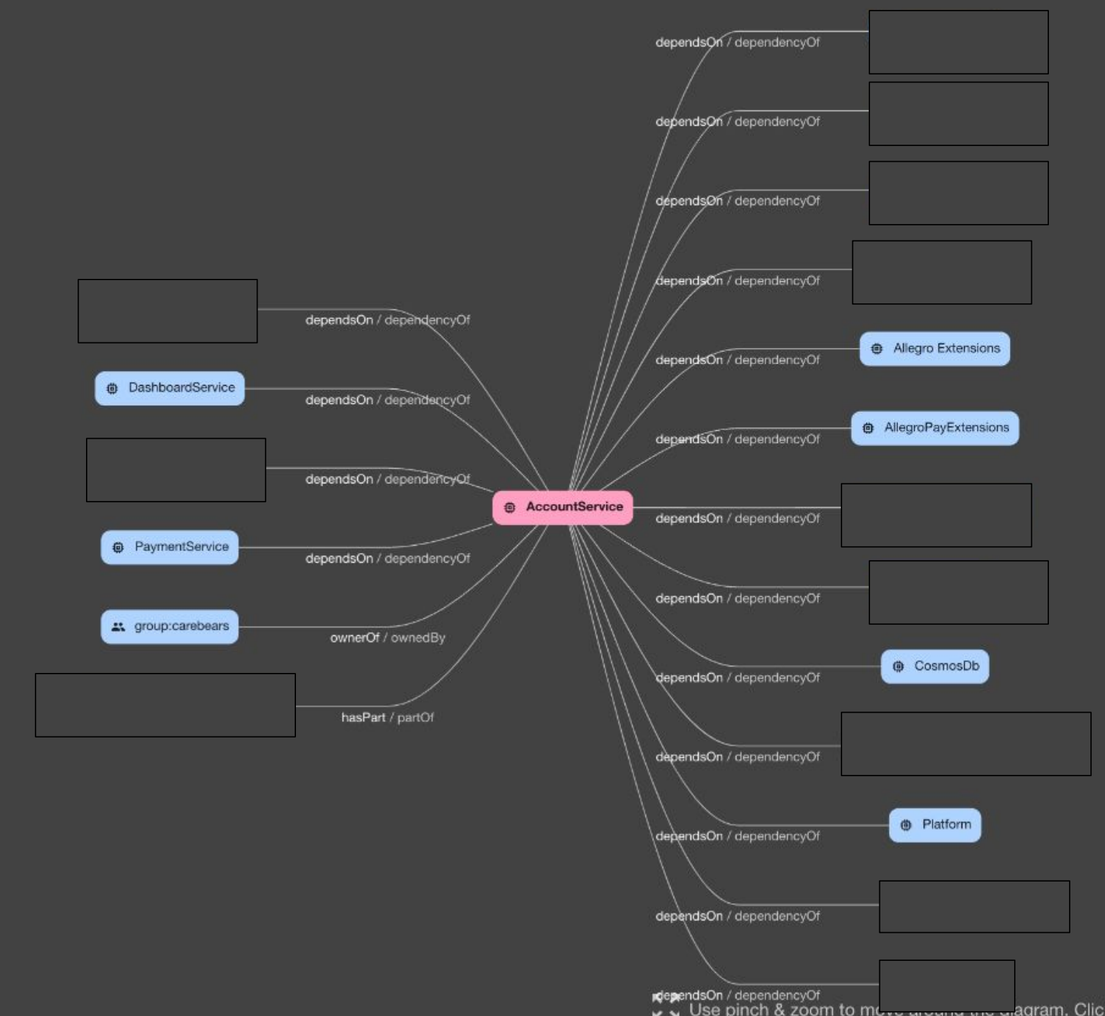
Left to right

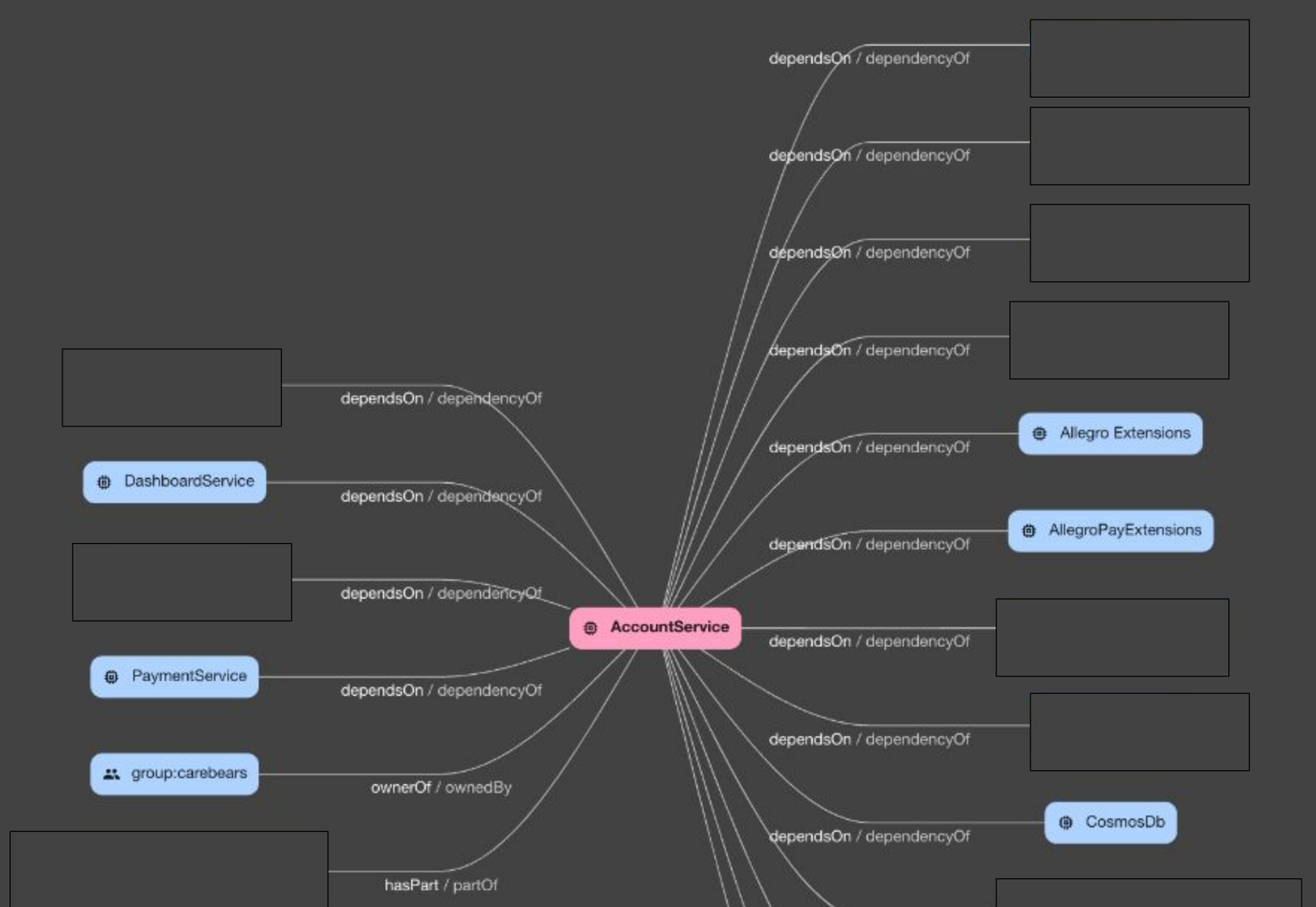
Curve

Monotone X

Simplified

Merge Relations




dependsOn / dependencyOf
dependsOn / dependencyOf
dependsOn / dependencyOf
dependsOn / dependencyOf
dependsOn / dependencyOf
Allegro Extensions
dependsOn / dependencyOf
AllegroPayExtensions
dependsOn / dependencyOf
dependsOn / dependencyOf
dependsOn / dependencyOf
CosmosDb
dependsOn / dependencyOf
dependsOn / dependencyOf
Platform
dependsOn / dependencyOf
dependsOn / dependencyOf
dependsOn / dependencyOf
DashboardService
dependsOn / dependencyOf
dependsOn / dependencyOf
AccountService
PaymentService
dependsOn / dependencyOf
group:carebears
ownerOf / ownedBy
hasPart / partOf

PagerDuty

SERVICE DIRECTORY | CREATE INCIDENT | ESCALATION POLICY

Incidents | Change Events

Nice! No incidents found!

ON CALL

Rafal Schmidt
rafal.schmidt@allegro.com

Allegro Pay Backstage docs

Component Backstage | **Owner** aard | **Lifecycle** experimental | Sou

Search Backstage docs

Allegro Pay Backstage docs

Home

Catalog entities

Catalog entities

Data in the catalog are gathered via integrations with:

- Azure DevOps (services, documentation, libraries, API, etc.)
- Azure AD (organization data)

Organization data

We have two main organization entity kinds:

- group (team)
- user

Only users that are in Azure Active Directory with the company name `Allegro Pay sp. z o.o.` are added to the catalog. Based on that users can log in with elevated permissions. Other users might access Backstage with a `guest` account.

Only listed teams are added via integration to the catalog, as there is no easy way to figure out what groups are teams in Allegro Pay. If you need to add a new team please go to #help-aard

Entities

The main entities kinds that we use are:

- component - service - web-site - library
- system

Table of contents

Organization data

Entities

Component

Common annotations

dev.azure.com/projec repo: <project-name>/<repo-name>

allegropay/entity-updated-at: '<yyyy-MM-ddThh:mm>'

Service & Website

azure.com/tag-selector: <tag-name>/<tag-value>

argocd/app-name: <argo-app-name>

backstage.io/kuberne label-selector: <aks-label-selector>

pagerduty.com/integr key: <pager-duty-integration-key>

Library

API

System

Service & Website

There are not many differences between `service` and `website` in case of functionalities other than purpose, so we describe them together.

Example:

```yaml
apiVersion: backstage.io/v1alpha1
kind: Component
metadata:
      name: entity-name
      description: entity-description
      title: entity-title
      annotations:
            dev.azure.com/project-repo: <project-name>/<repo-name>
            azure.com/tag-selector: <tag-name>/<tag-value>
            backstage.io/techdocs-ref: dir:.
            argocd/app-name: <argo-app-name>
            backstage.io/kubernetes-namespace: <aks-namespace>
            backstage.io/kubernetes-label-selector: <label>=<label-value>
            allegropay/entity-updated-at: '<yyyy-MM-ddThh:mm>'
            pagerduty.com/integration-key: <pager-duty-integration-key>
      links:
            - url: https://backstage.westeurope.dev.devops.fintech.allegrogroup.com/
              title: Backstage Dev
              icon: dashboard
              type: website
            - url: https://backstage.westeurope.xyz.devops.fintech.allegrogroup.com/
              title: Backstage Xyz
              icon: dashboard
              type: website
            - url: https://dev.azure.com/AllegroTechies/AllegroPay.DevOps/_git/ops-k8s-infrastructure?path=/backstage
              title: Backstage Argo Helm Chart repository
              icon: github
              type: repository
spec:
      type: website | service
      owner: team-name
      lifecycle: production | experimental | deprecated
      consumesApis:
            - consumed-api
      providesApis:
            - entity-name-api
```

```
spec:
      type: website | service
      owner: team-name
      lifecycle: production | experimental | deprecated
      consumesApis:
            - consumed-api
      providesApis:
            - entity-name-api
```

```
3  metadata:
4        name: entity-name
5        description: entity-description
6        title: entity-title
7        annotations:
8              dev.azure.com/project-repo: <project-name>/<repo-name>
9              azure.com/tag-selector: <tag-name>/<tag-value>
10             backstage.io/techdocs-ref: dir:.
11             argocd/app-name: <argo-app-name>
12             backstage.io/kubernetes-namespace: <aks-namespace>
13             backstage.io/kubernetes-label-selector: <label>=<label-value>
14             allegropay/entity-updated-at: '<yyyy-MM-ddThh:mm>'
15             pagerduty.com/integration-key: <pager-duty-integration-key>
```

Sourcegraph

Code search and an AI assistant with the context of the code graph.

context:global Search for code or files... Aa .* []

I can help with that.

Meet Cody, your AI assistant

Install Cody for your IDE

Cody for your IDE provides the power of LLMs to help you generate and fix code, right where you commit.

View editor extensions →

Cody for Sourcegraph.com

A free, helpful AI assistant, that explains, generates, and transpiles code, in the Sourcegraph web interface.

Try Cody chat →

Try Cody on a file →

Introduction to Batch Changes

Overview

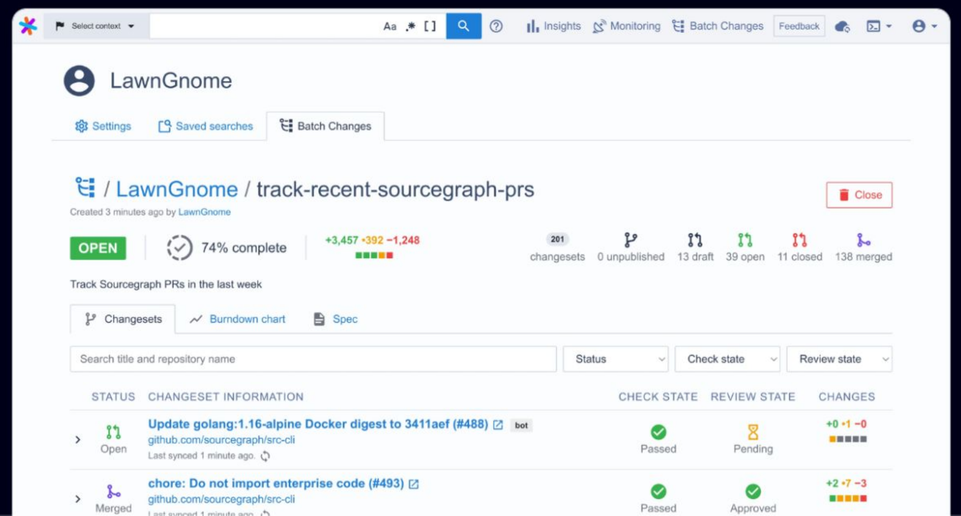
Batch Changes let you make large-scale code changes across many repositories and code hosts. Batch Change lets you create pull requests on all affected repositories, and it tracks their progress until they're all merged. You can preview the changes and update them at any time.

People usually use Batch Changes to make the following kinds of changes:

- Cleaning up common problems using linters.
- Updating uses of deprecated library APIs.
- Upgrading dependencies.
- Patching critical security issues.
- Standardizing build, configuration, and deployment files.

A batch change tracks all of its changesets (a generic term for pull requests or merge requests) for updates to:

- Status: open, merged, or closed
- Checks: passed (green), failed (red), or pending (yellow)
- Review status: approved, changes requested, pending, or other statuses (depending on your code host or code review tool)



Search in Allegro Pay Backstage

CLEAR

```
<type>[optional scope]: <description>

[optional body]

[optional footer]
```

Change Log

All notable changes to this project will be documented in this file. See standard-version for commit guidelines.

1.2.1 (2019-04-25)

Bug Fixes

- typo in readme (ba3bd01)

1.2.0 (2019-04-25)

Bug Fixes

- crash on connection reset (43957f3)

Features

- docker (804017a)
- readme (398a63c)

1.1.0 (2019-04-25)

Features

- first implementation of telnet server e7a98a5

Change Log

All notable changes to this project will be documented in this file. See standard-version for commit guidelines.

1.2.1 (2019-04-25)

Bug Fixes

- typo in readme (ba3bd01)

1.2.0 (2019-04-25)

Bug Fixes

- crash on connection reset (43957f3)

Features

- docker (804017a)
- readme (398a63c)

1.1.0 (2019-04-25)

Features

- first implementation of telnet server e7a98a5

README.md

Conventional Changelog

build passing coverage 94%

Generate a CHANGELOG from git metadata.

About this Repo

The conventional-changelog repo is managed as a monorepo; it's composed of many npm

The original `conventional-changelog/conventional-changelog` API repo can be found in changelog.

Getting started

It's recommended you use the high level standard-version library, which is a drop-in replac `version` command, handling automated version bumping, tagging and CHANGELOG gen

Alternatively, if you'd like to move towards completely automating your release process as an output from CI/CD, consider using semantic-release.

You can also use one of the plugins if you are already using the tool:


REST
Client

example.http

```
GET https://api.github.com/users/Huachao

###

GET https://api.github.com/repos/Huachao/vscode-restclient HTTP/1.1

###

POST http://requestb.in/tcdce4tc?k=a%20bc
Date: {{$guid}}

{
    "key": "{{$guid}}",
    "key2": "{{$timestamp -1 d}}",
    "key3": "{{$guid}}",
    "key4": "{{$Guid}}"
    #rest
    "key5": "{{$timestamfasdfa}}",
    "key6": "{{}}",
    "key7": "{{d}}",
    "key8": "{{$timestamp}}",
    "key9": "{{$randomInt 1 19}}"
    "key10": "{{$randomInt 21 19}}",
}
```

1. Target groups - “who”

1. Target groups - “who”
2. Use cases - “what”

1. Target groups - “who”
2. Use cases - “what”
3. Tooling - “how”

MenziesTheHeretic · 5 yr. ago

- making the parent dirty
- marking a single child as touched
- killing first child, then spawning a new one

Thank god for the garbage collector

Thank you!

Got any questions?

rafal.schmidt@allegro.com
linkedin.com/in/rafalschmidt

Jakie narzędzia pomagają w budowaniu dokumentacji?

23.01.2024

Co dokumentujemy, a co powinniśmy.

22.01.2024

Dla kogo piszemy dokumentację?

03.01.2024

rafalschmidt.com

Thank you!

Got any questions?

rafal.schmidt@allegro.com
linkedin.com/in/rafalschmidt