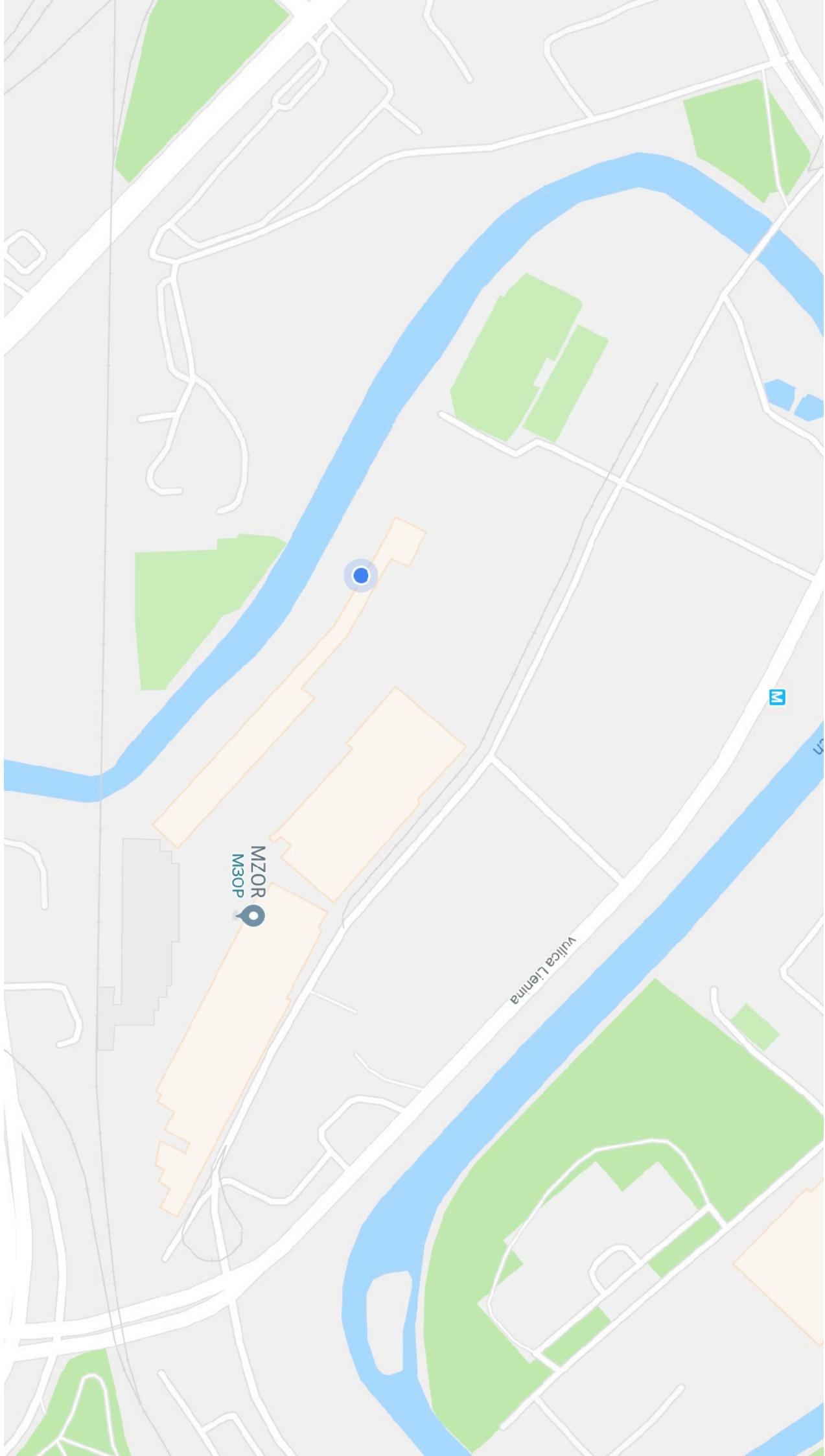


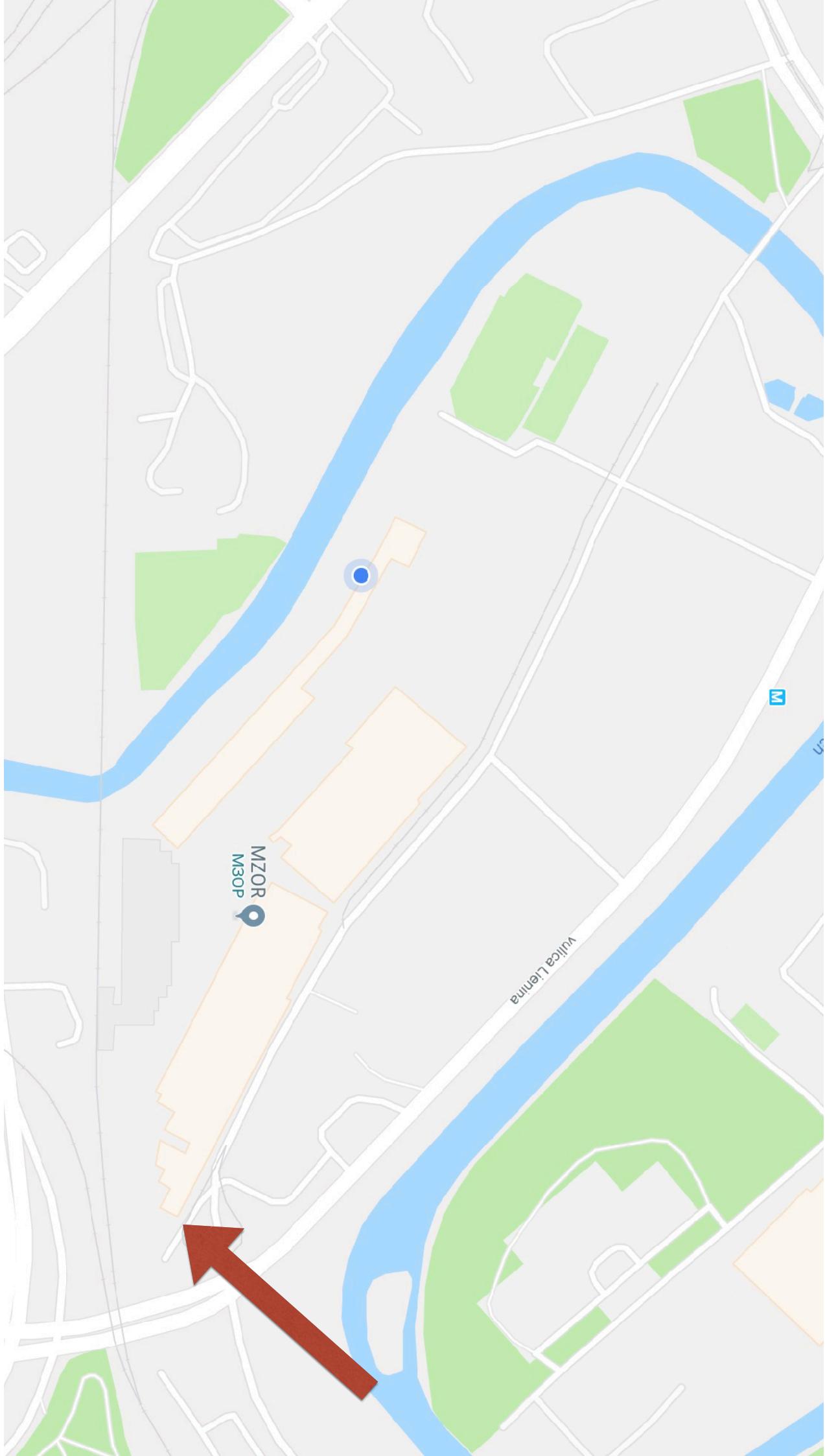
Hello Tver!



My first time here

Hard weeks



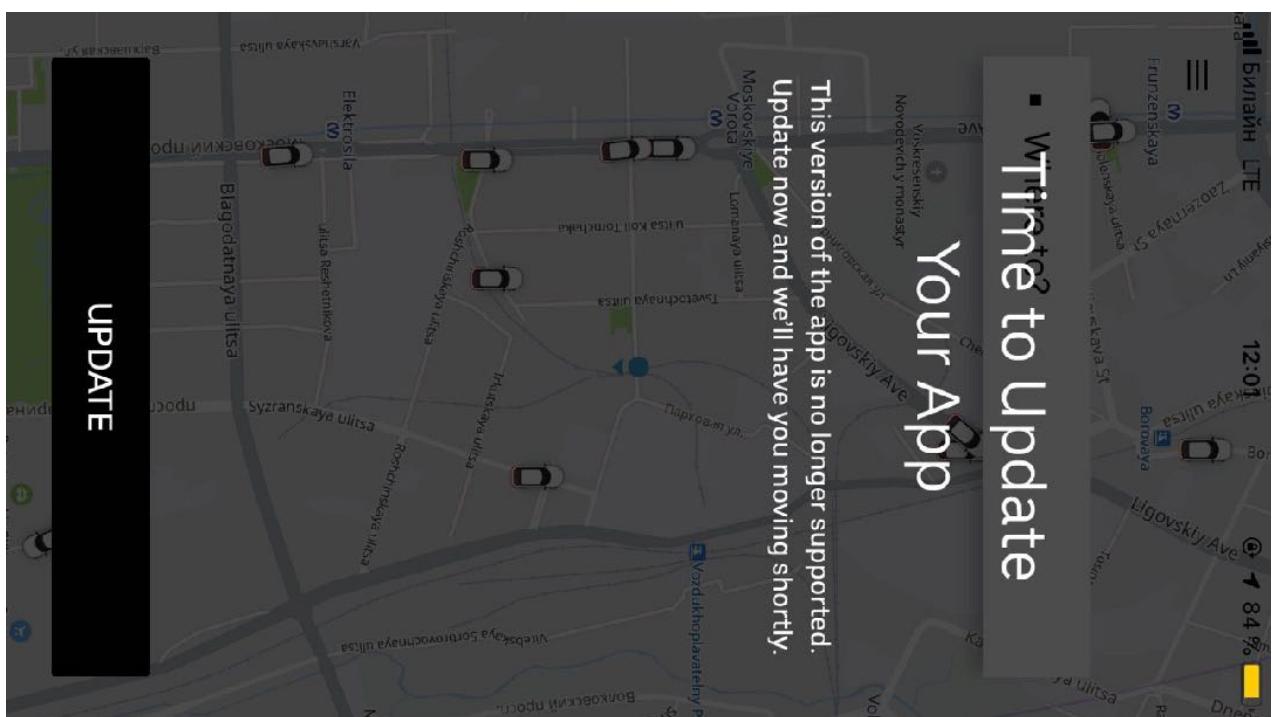




MZOR
M3OP

Vileta Lenina

Завтра Выступать 04:12 //



Билайн LTE

11:30

④ 7 94 %

X

Enter your phone number

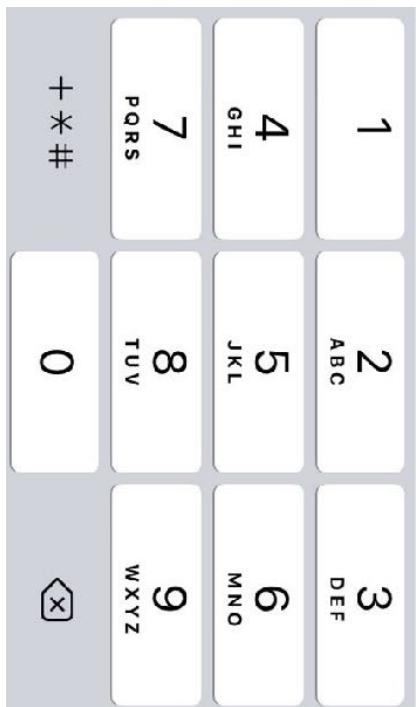
Russia +7 ▾

+7 968

Sent SMS limit exceeded

NEXT

By tapping "Next", I accept the terms of the
[License Agreement](#)





Enter your phone number

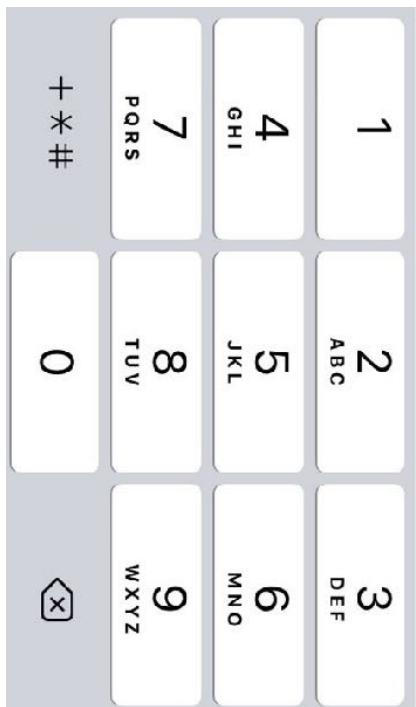
Russia +7 ▾

+7 968

Sent SMS limit exceeded

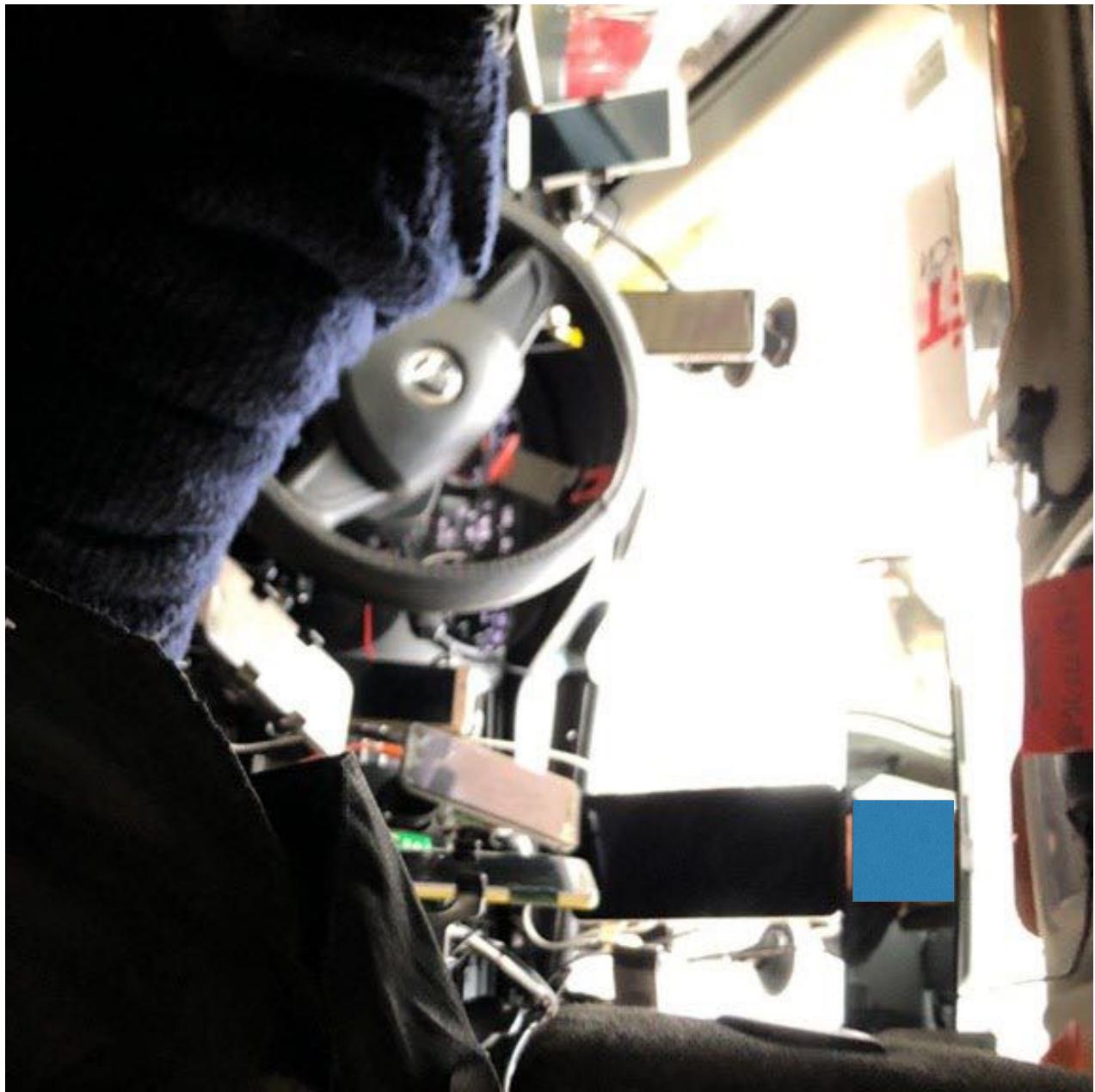
By tapping "Next", I accept the terms of the
[License Agreement](#)

Next



DRIVEN BY QUALITY





[NEW COMMENTS](#)

Winter SaintP Ruby Meetup

with Saint-P Ruby User Group



Dmitry Tigrov

Одна кофемашина на 100 человек не
серъезно

[Like](#) · [Reply](#)

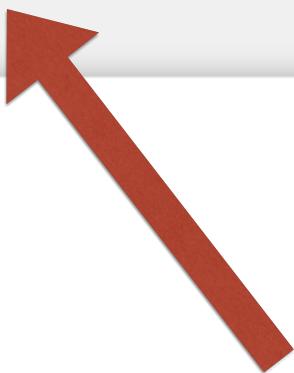
11:17 AM

[View conversation on Meetup](#)

274

Thank you ❤

github.com/davydovanton
twitter.com/anton_davydov
davydovanton.com







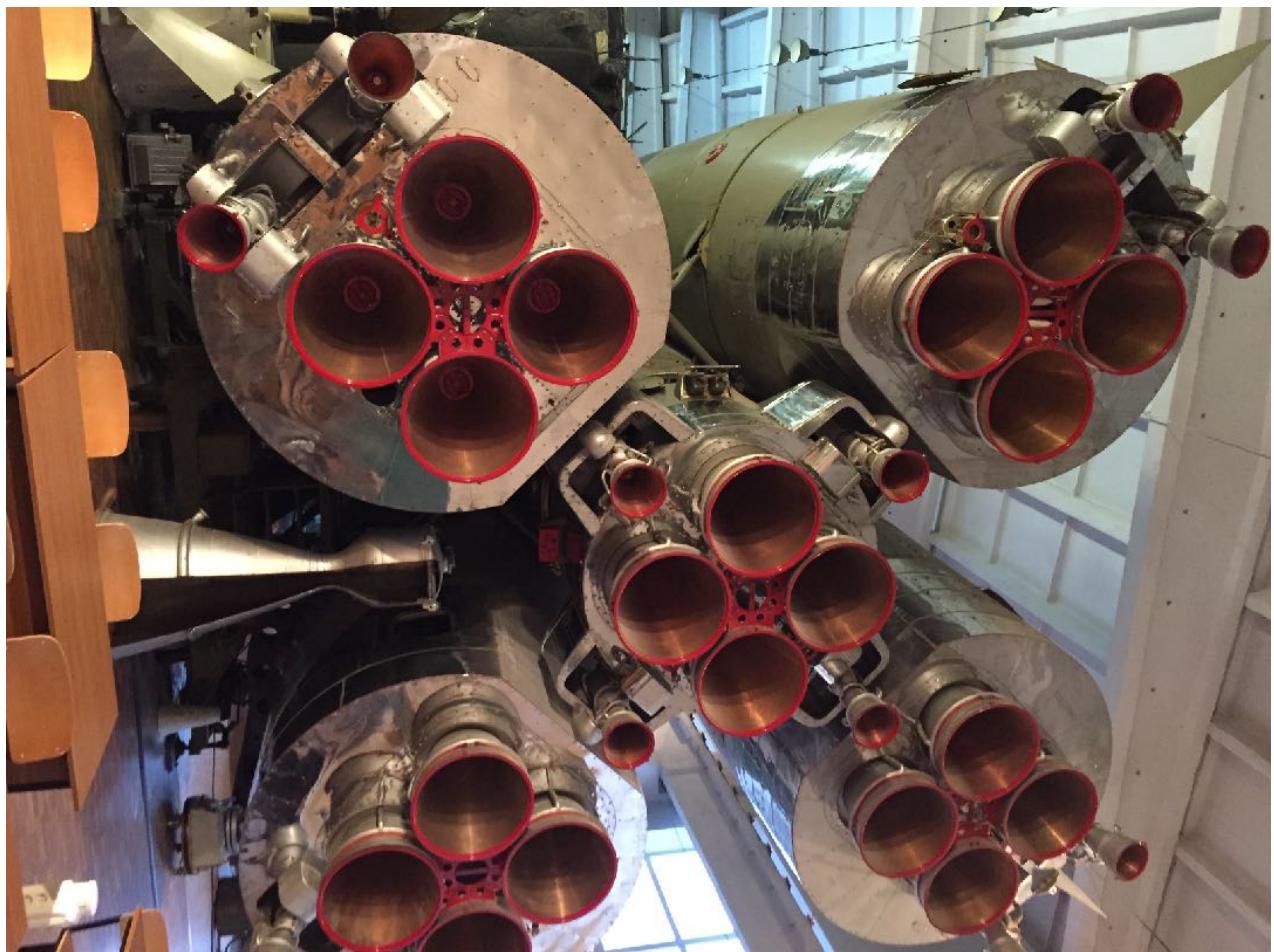
古事記の世界

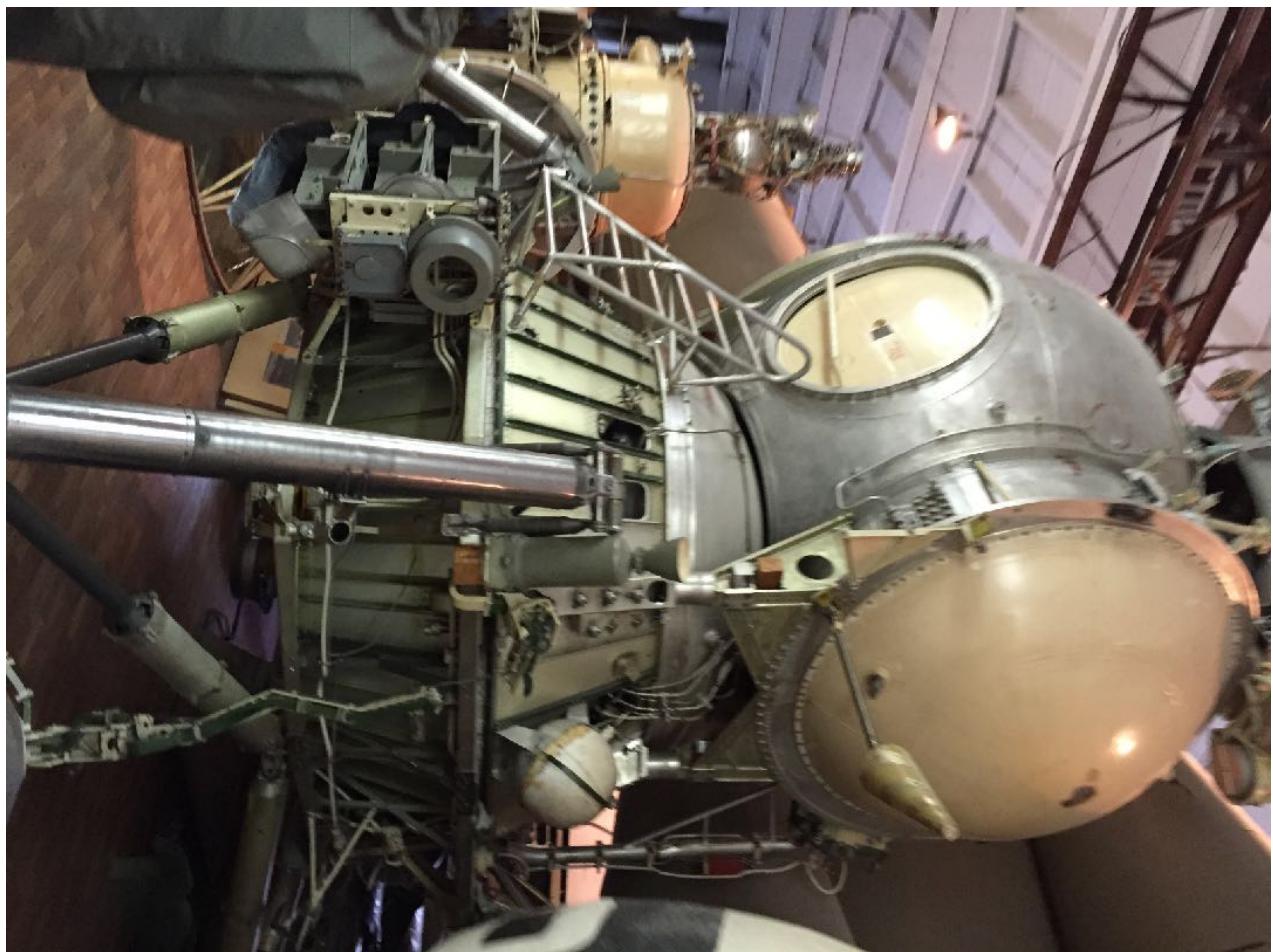
力の傳の子アーヴィング











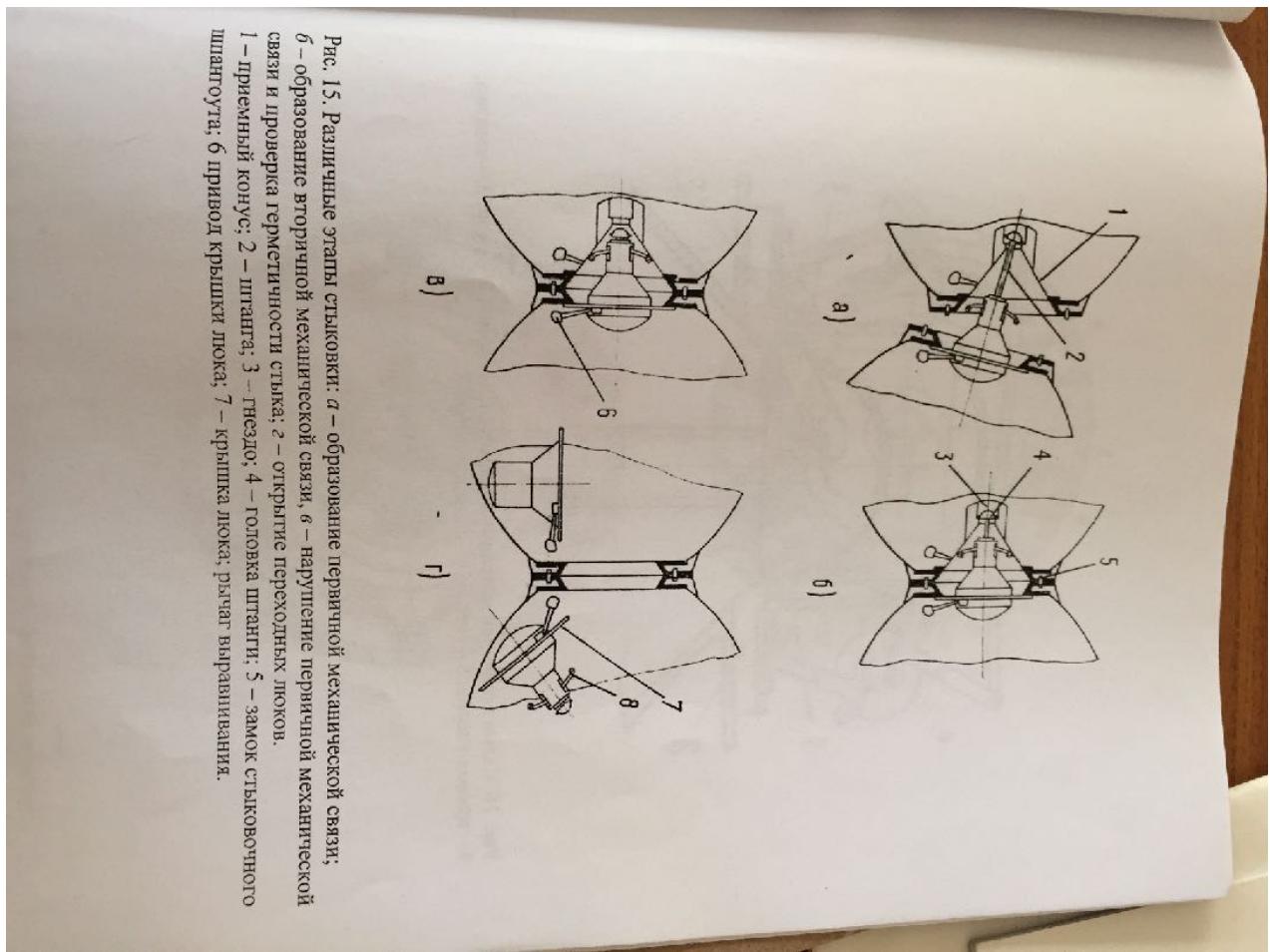
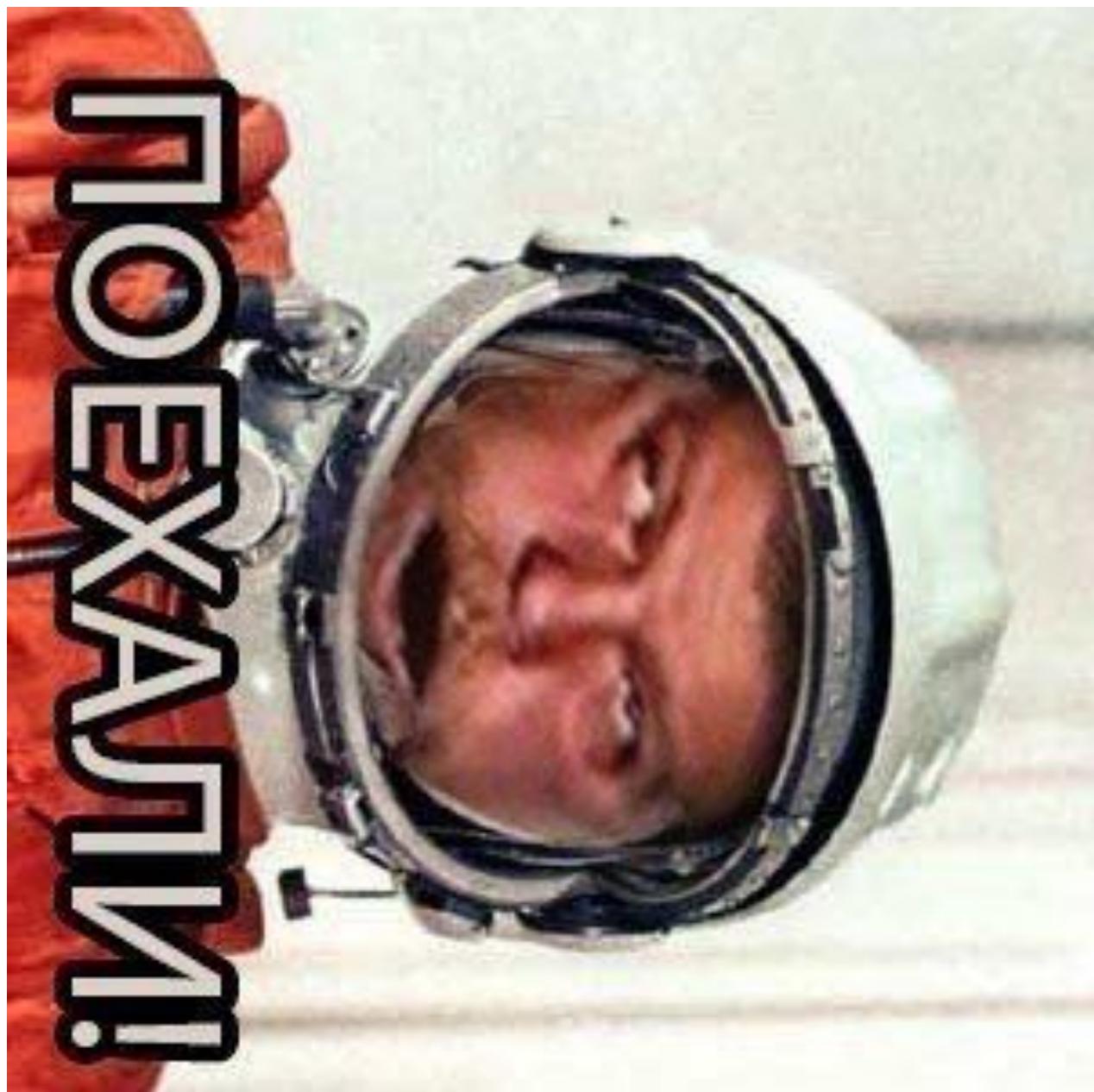


Рис. 15. Различные этапыстыковки: а – образование первичной механической связи; б – образование вторичной механической связи; в – нарушение первичной механической связи и проверка герметичностистыков; г – открытие переходных локов. 1 – приемный конус; 2 – панта; 3 – гнездо; 4 – головка панти; 5 – замок стыковочного шлангута; 6 – привод крышки люка; 7 – крышка люка, рычаг выравнивания.

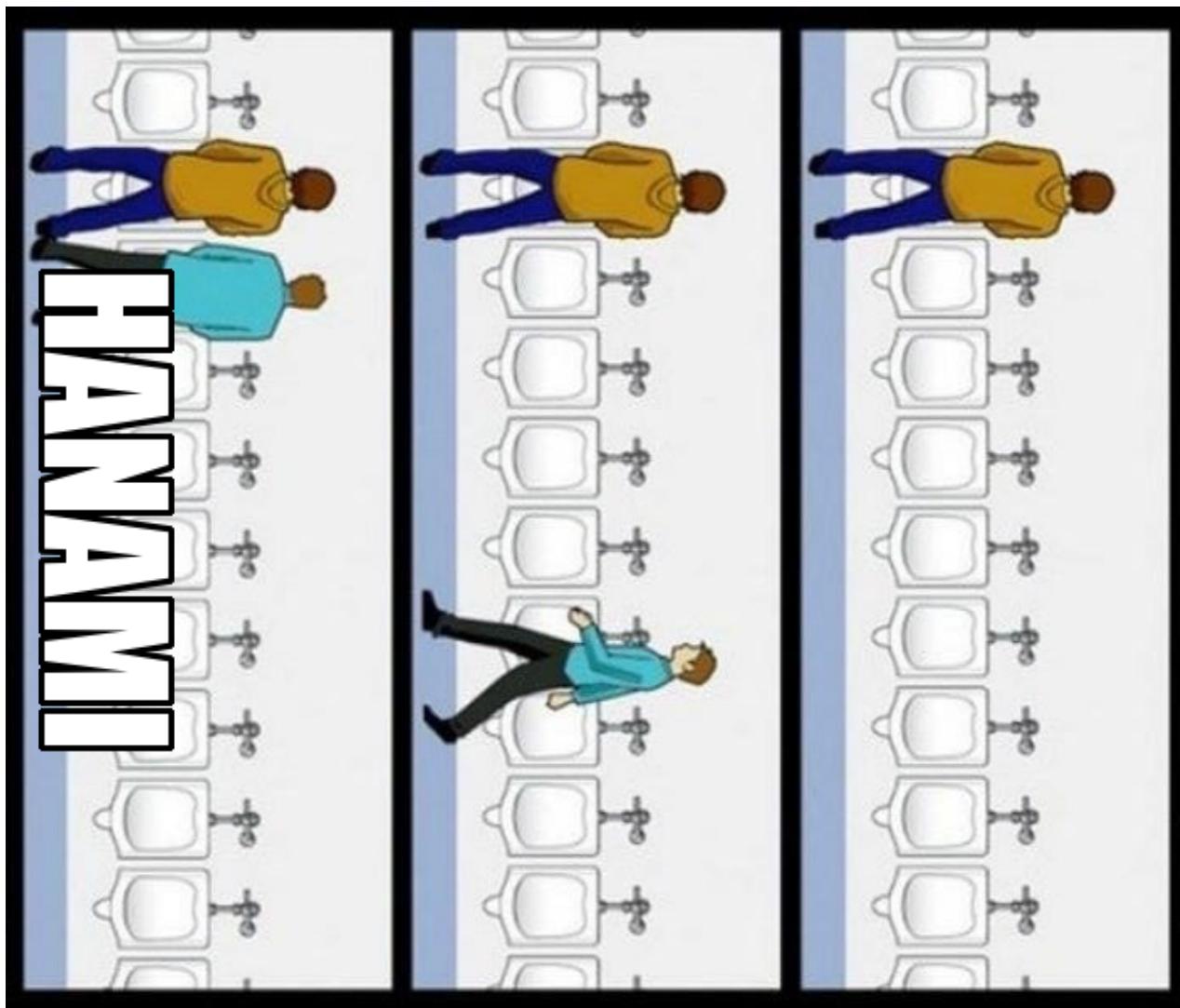


ONE
WAX



Anton Davydov

github.com/davydovanton
twitter.com/anton_davydov
davydovanton.com



- Software developer at Hippo
- OpenSource evangelist
- Hanami core developer
- Try to make own product
- Technical consulting



2|rogramming

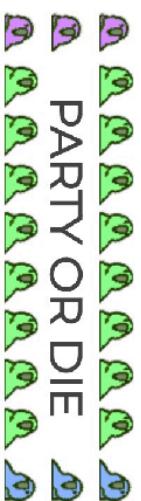
777 subscribers

info

Грустно о программировании. Все проблемы сюда: [@davydovanton](https://t.me/davydovanton)

twitch.tv/davydovanton

Cult of the Party Parrot



- | | | | |
|------------------------------------|---------------------------------------|------------------------------------|----------------------------|
| Parrot ¹⁰ | Open-source Parrot ¹⁰ | Middle Parrot ¹⁰ | Right Parrot ¹⁰ |
| Aussie Parrot | Old Timey Parrot | Bored Parrot | |
| Shuffle Parrot ¹⁰ | Conga Line Parrot ^{10*} | Reverse Conga Line Parrot | |
| Party Parrot ¹⁰ | Parrot Cop ^{10*} | Fast Parrot ^{10*} | |
| Ultra Fast Parrot ^{10*} * | Slo-mo Parrot* | Parrot Dad* | |
| Deal With It Parrot ¹⁰ | Deal With It Now Parrot ¹⁰ | Explody Parrot | |
| Parrot | Fiesta Parrot | Confused Parrot ¹⁰ | |
| Shuffle Party Parrot | Ice Cream Parrot | Parrot Wave (1) | |
| Aussie Conga Line Parrot | Aussie Reverse Conga Line Parrot | Parrot Wave (2) | |
| Parrot Wave (3) | Parrot Wave (4) | Parrot Wave (5) | |
| Parrot Wave (7) | Parrot Wave (8) | Parrot Wave (9) | |
| Moonwalking Parrot | Thumbs Up Parrot ¹⁰ | Parrot with mustache ¹⁰ | |
| Christmas Parrot ¹⁰ | Sleepy Parrot | Dark Beer Parrot | |
| Blonde Sassy Parrot | Blues Clues Parrot | Margarita Parrot* | |
| Dreidel Parrot* | Harry Potter Parrot* | Twins Parrot ^{10*} | |



Anton Davyдов

я все же считаю, что party parrot самая киппер фича слака

<https://cultofthepartyparrot.com>

Cult of the Party Parrot

Cult of the Party Parrot if the Party if
THE PARROT INSISTS



Today

Screen Shot 2019-01-21 at 20.10.55.png

 Anton Davydov
я все же считаю, что party parrot самая киллер фича спака

Cult of the Party Parrot THE PARROT INSISTS *of the Party*

A vertical column of 15 numbered boxes, each containing a different fruit or vegetable icon. The icons include: 1. Tomato, 2. Onion, 3. Carrot, 4. Potato, 5. Cucumber, 6. Bell Pepper, 7. Tomato, 8. Onion, 9. Carrot, 10. Potato, 11. Cucumber, 12. Bell Pepper, 13. Tomato, 14. Onion, 15. Carrot.

rodin 21:15

(edited)

4

я все же считаю

Cult of the Party Parrot
Cult of the Party Parrot

• **PARTY ON**

Posted in #random | Today at 21:12 | View message

2

21:16

1

上

new messages



Open
Source



- Coffee
- Beer
- Psychology
- Nintendo Switch:

SW-2039-0208-4228

—
love stories

Why I'm here?

Share experience

Suggest and discuss solutions

To take a beer on the afterparty

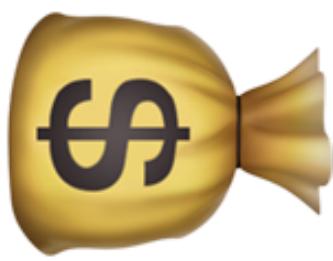




Inception

E-commerce

Big system with small
sub systems



Prices

Young project

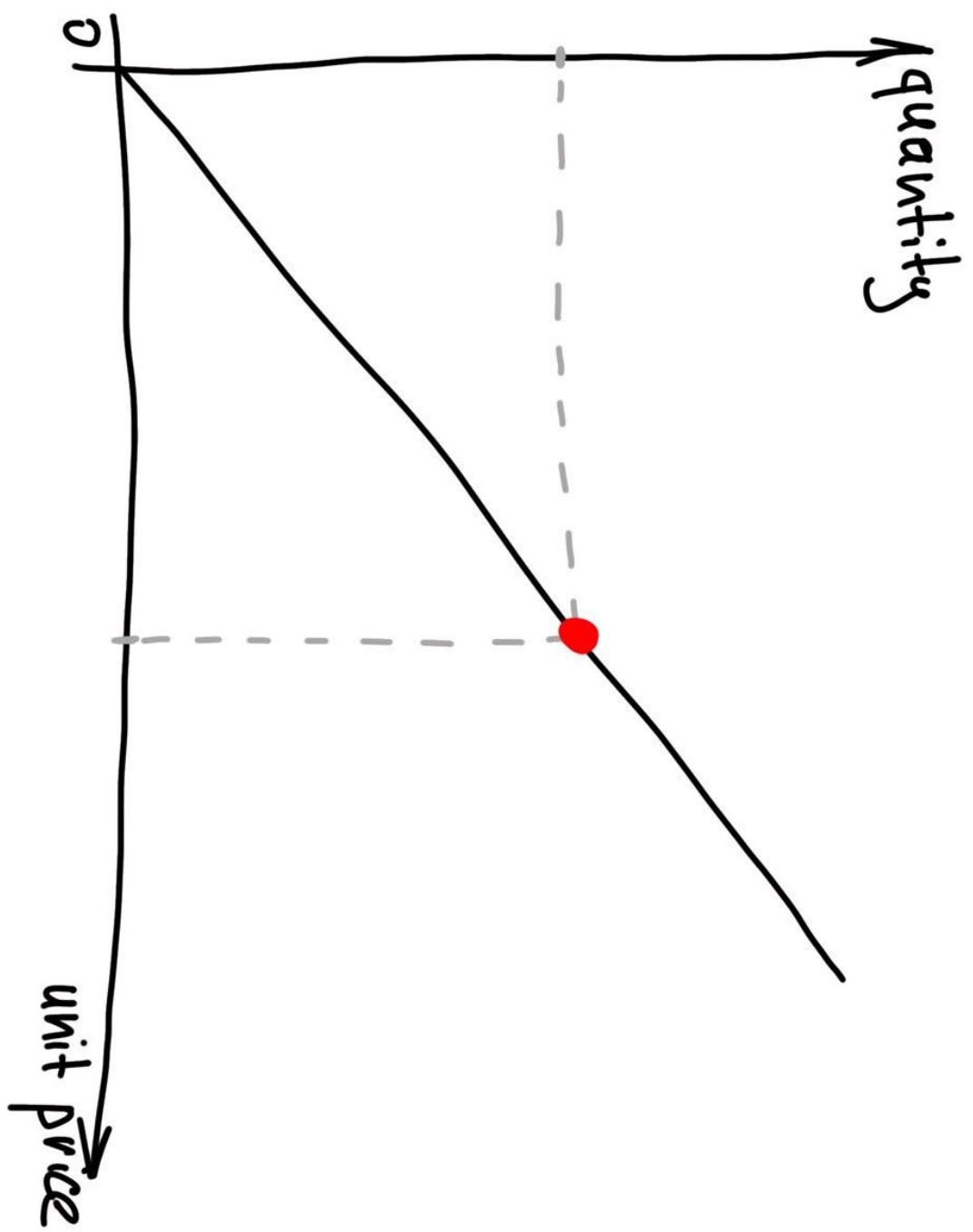
Simple price calculation

total price = unit price * quantity

All logic in the model in
monolith app

```
module Entities
  class Price
    def cost_for(quantity)
      unit_price * quantity
    end
  end
end
```

Linear Calculation





Complicating

Business understand that we
need to use fees and margins
for getting more money

Dynamic and static
for specific products

Examples

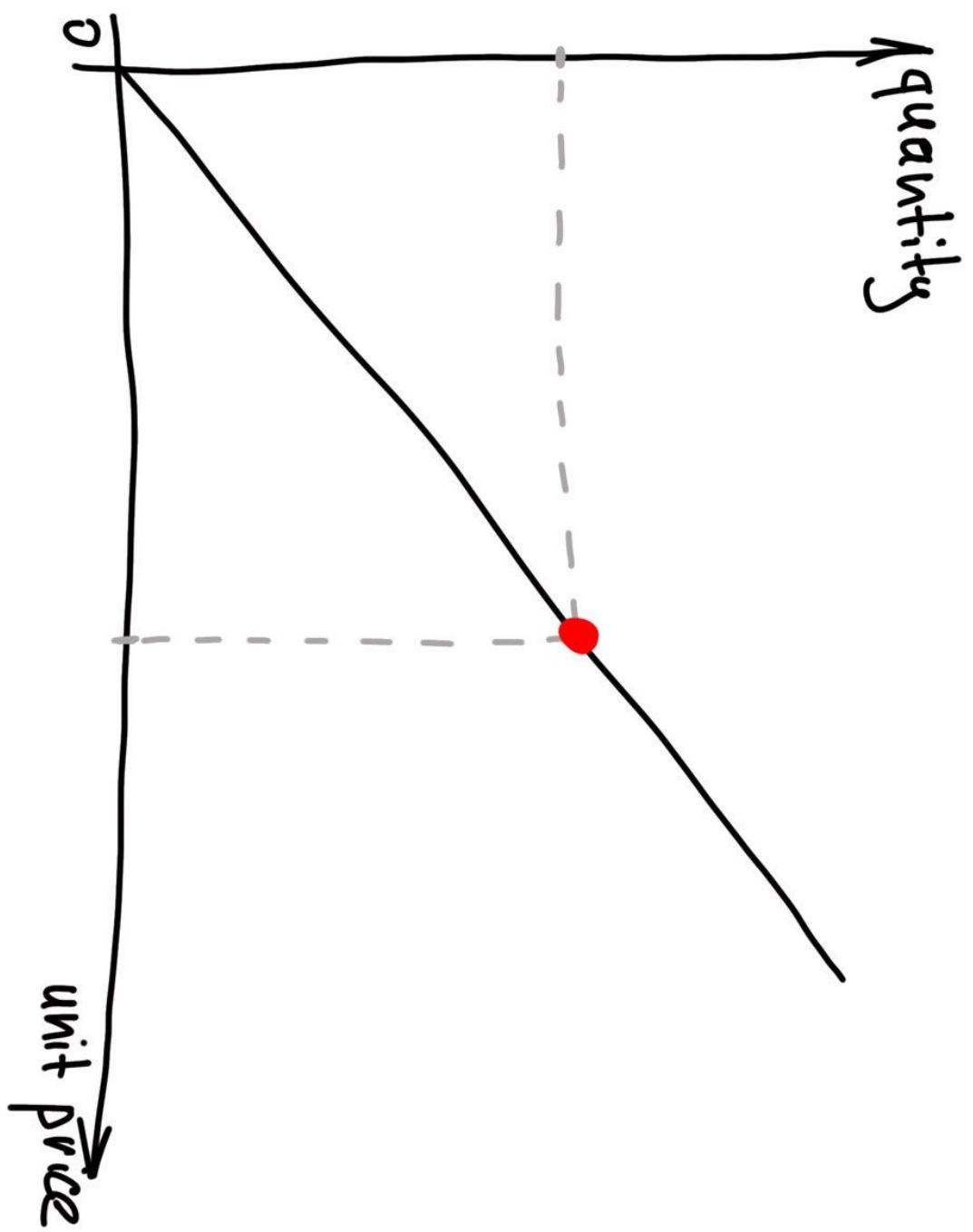
Static: payment processing fee

Dynamic: brand related margin

Calculation logic need to be
more complicated

```
module Entities
  class Price
    def cost_for(quantity)
      PriceCalculator.new.call(self, quantity)
    end
  end
end
```

Still linear calculation



3

Humans

Managers understand that
they can update prices
manually

For this we need UI/UX

(dashboard)

Two ways for setup

prices

#1

Setup price for units,
calculate margins

#2

Setup margins, calculate
price for units

Hanami
△³

```
module Entities
  class Price
    def cost_for(quantity)
      if total_price
        (total_price / (30 * DIMENSION)) * quantity
      else
        PriceCalculator.new.call(self, quantity)
      end
    end
  end
end
```

Problems

It was a prototype

Worked on it alone

Complicated UI



But it works and
everyone was happy

4

MicroService

Current problems

Price calculation logic in 3 different places

High cost for changes

Data scientist updates
data by SQL



data scientist

All Images Videos News Shopping More Settings

resume

meme

flat icon

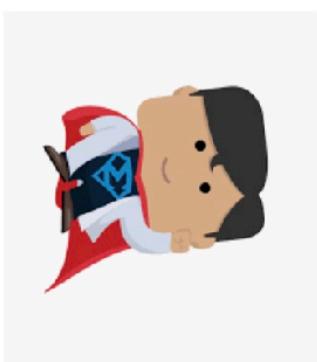
analyst professional

data analyst

resume sample



If you want to be data-driven, embrace ...
diginomica.com



A key data analytics role and a...
clusterdatal.nl



Different Career Paths in Data Sci...
slideshare.net

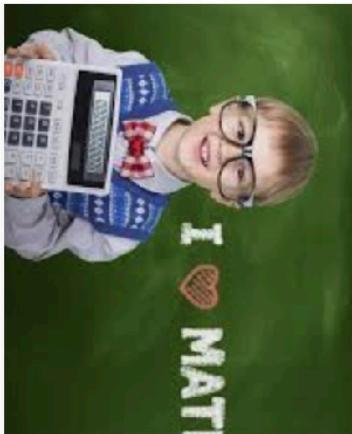


Data Scientist the Most Coveted Job ...

optymyze.com



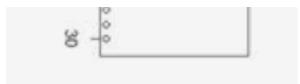
Lifetime Lessons: 20 Things Every ...
analyticsvidhya.com



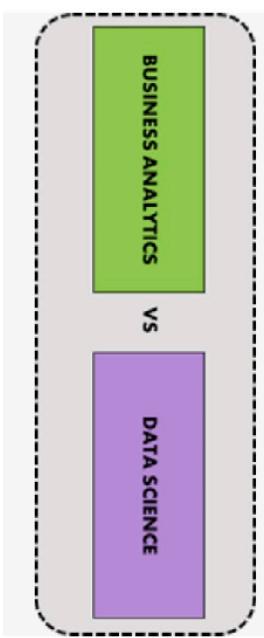
Data Scientist the Most Coveted Job ...

optymyze.com

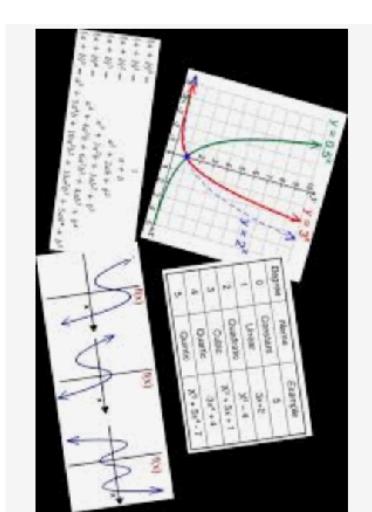
Sean McClure's Blog - Data Science C...
datasciencecentral.com



30

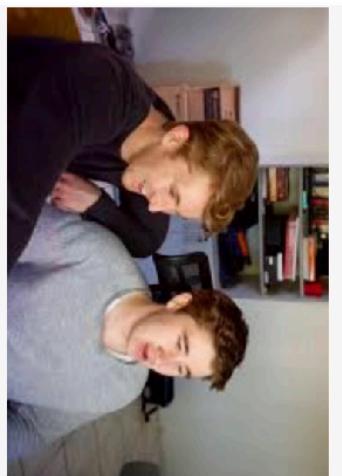


Essential Math for Data Science ...
[medium.com](https://medium.com/@mrbourke/essential-math-for-data-science-101-1000e3a2a2)



10

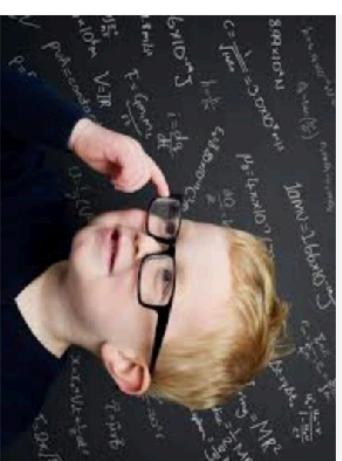
Blog — Daniel Bourke
mrbourke.com



Kaveh Bakhtiyari - Artificial ...
baktiyari.com



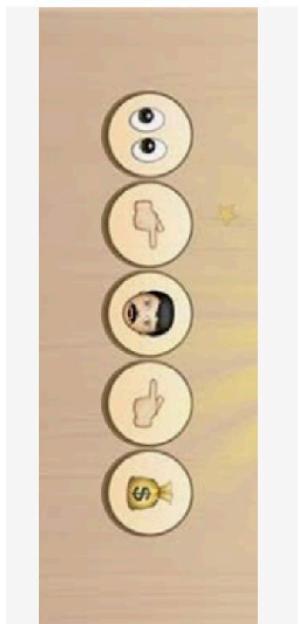
Lin(Luke) Qiu | LinkedIn
[linkedin.com](https://www.linkedin.com/in/lqiu/)

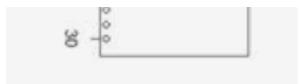


It's ideology vs. science in ...
nationalpost.com

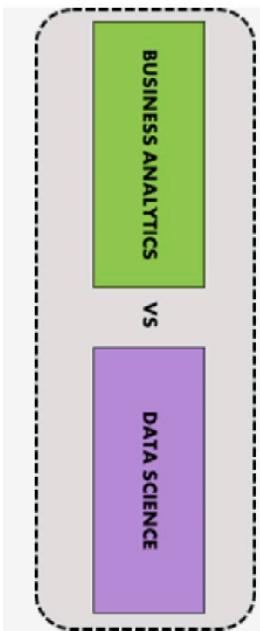


10 Popular TV Shows on Data Science and ...
analyticsvidhya.com

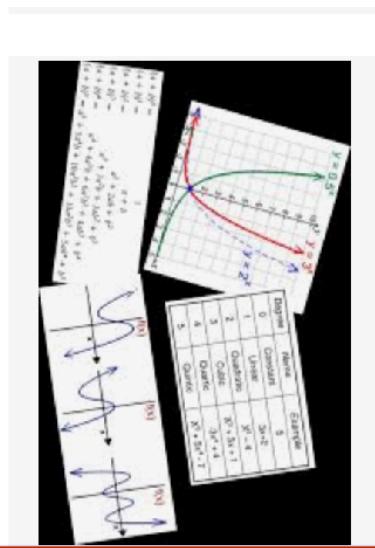




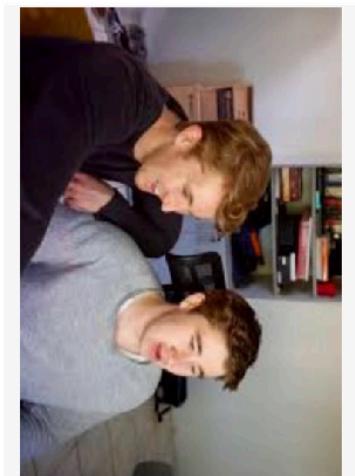
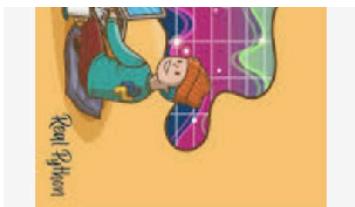
30



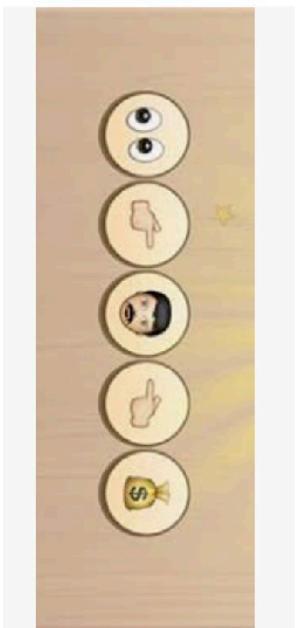
Essential Math for Data Science ...
[medium.com](https://medium.com/@mrbourke/essential-math-for-data-science-101-1000e3a2a2)



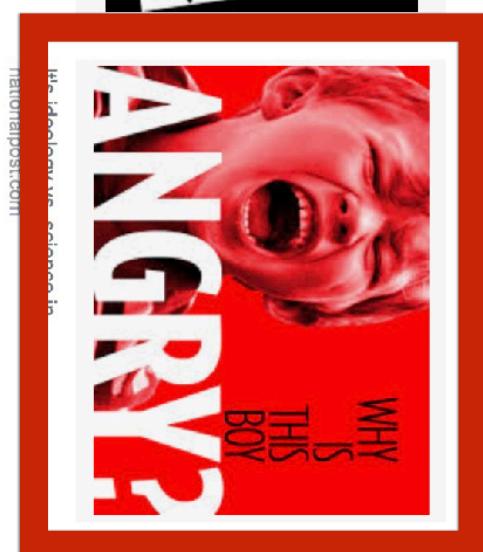
Ton
Blog — Daniel Bourke
mrbourke.com



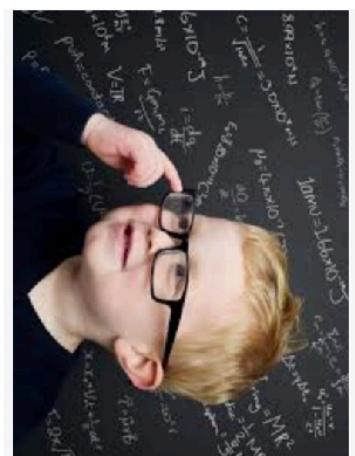
Kaveh Bakhtiyari - Artificial ...
baktiyari.com



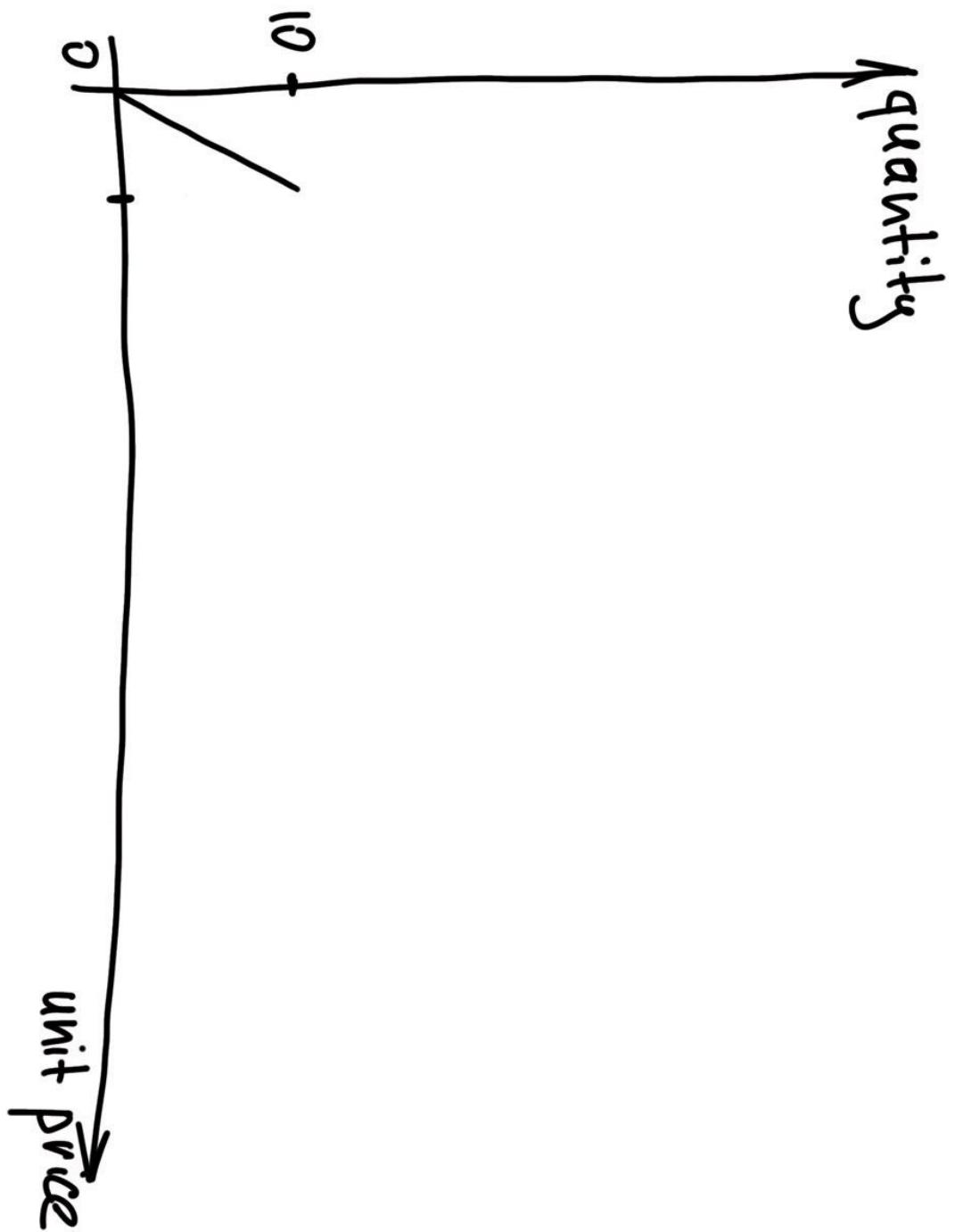
10 Popular TV Shows on Data Science and ...
analyticsvidhya.com

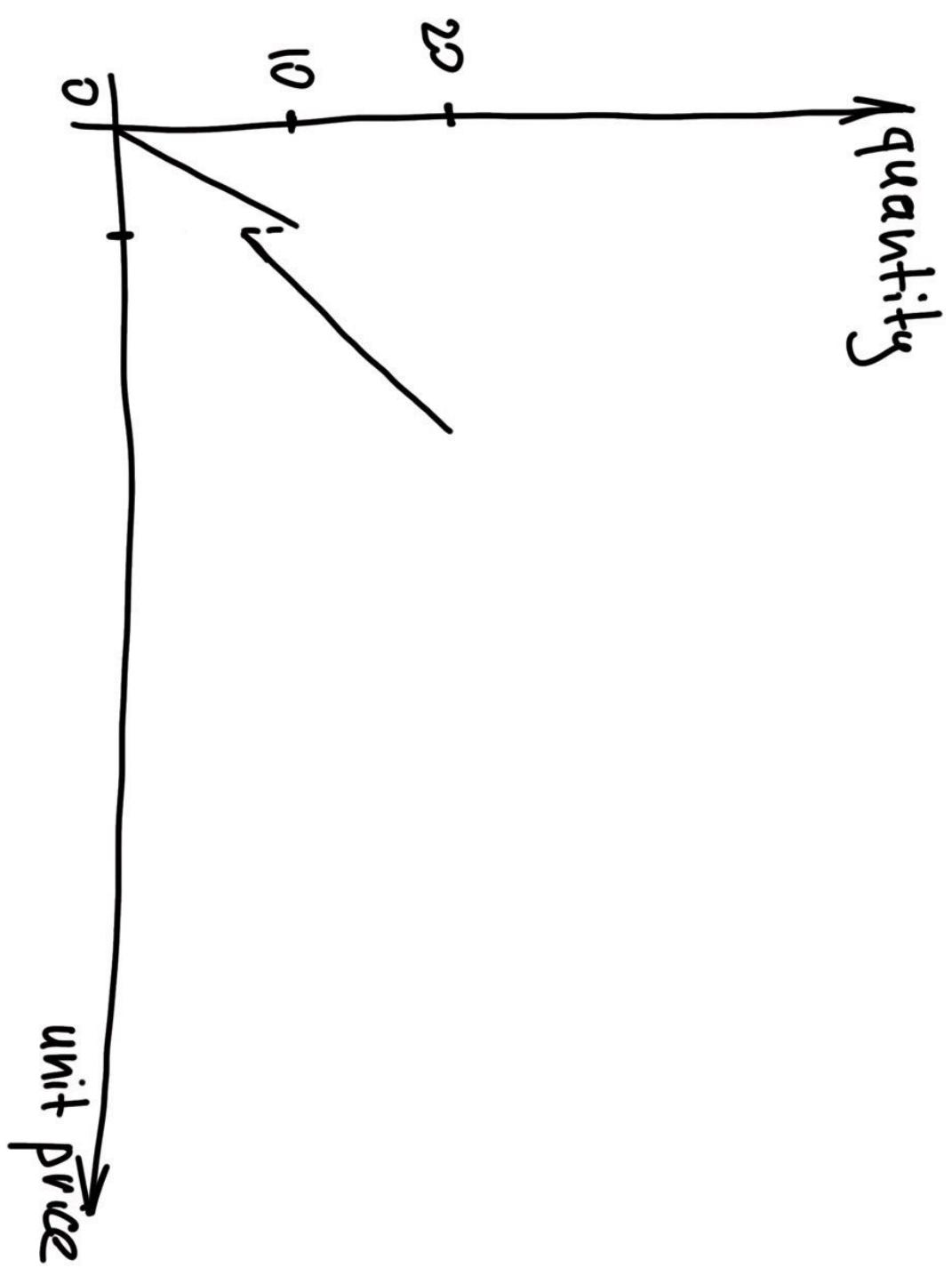


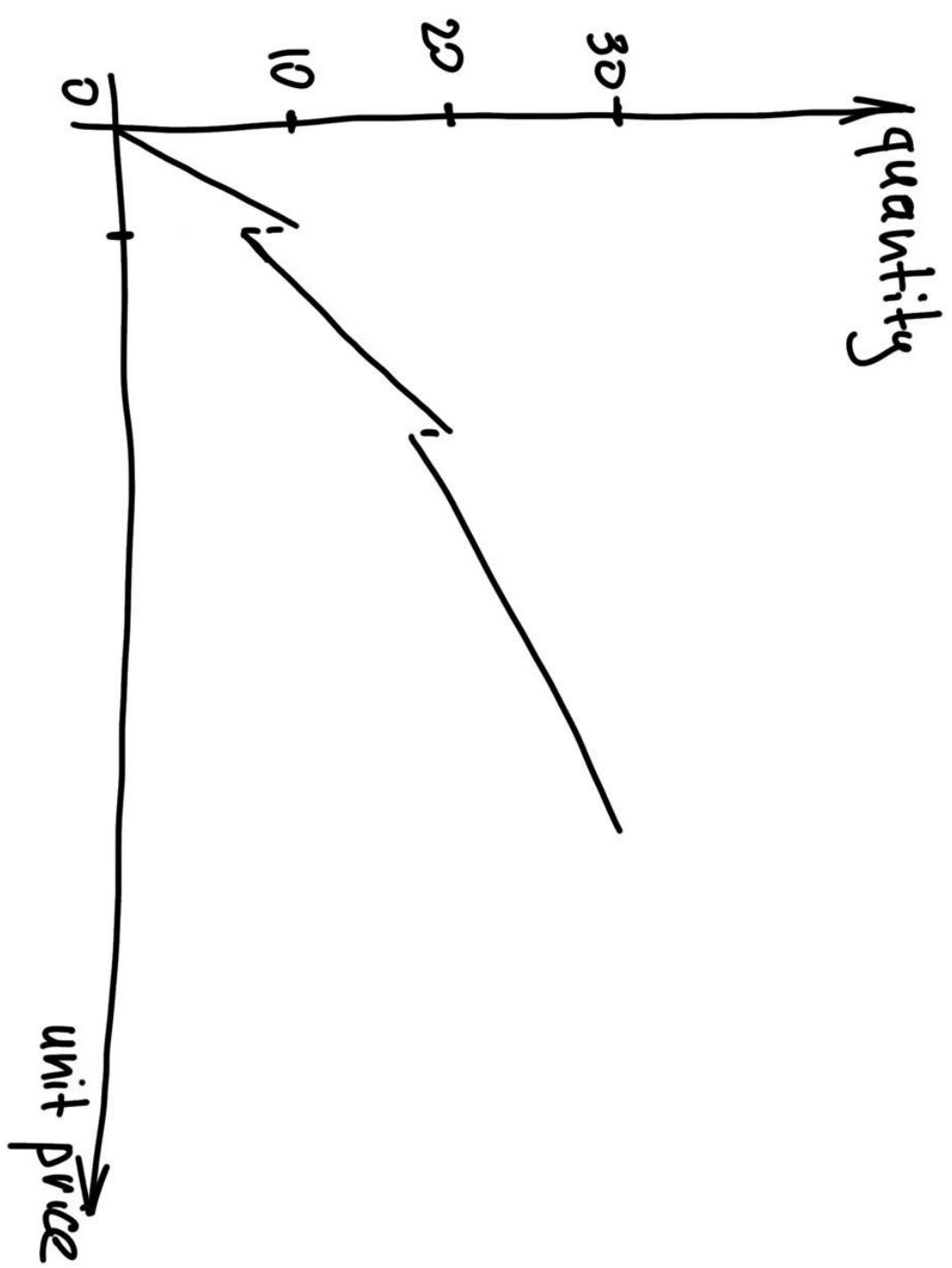
Lin(Luke) Qiu | LinkedIn
[linkedin.com](https://www.linkedin.com/in/lqiu/)



Not linear prices logic







Preparation

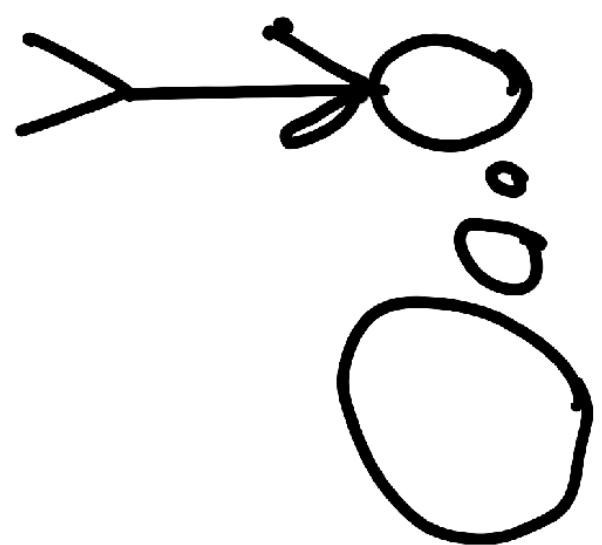
No requirements, only problems

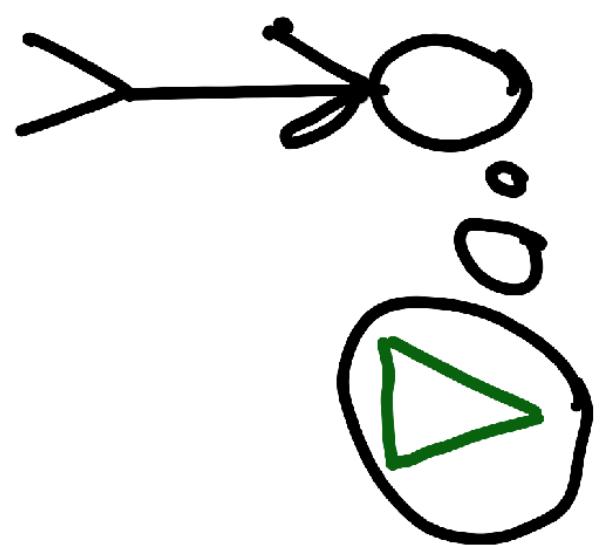
What I did for this

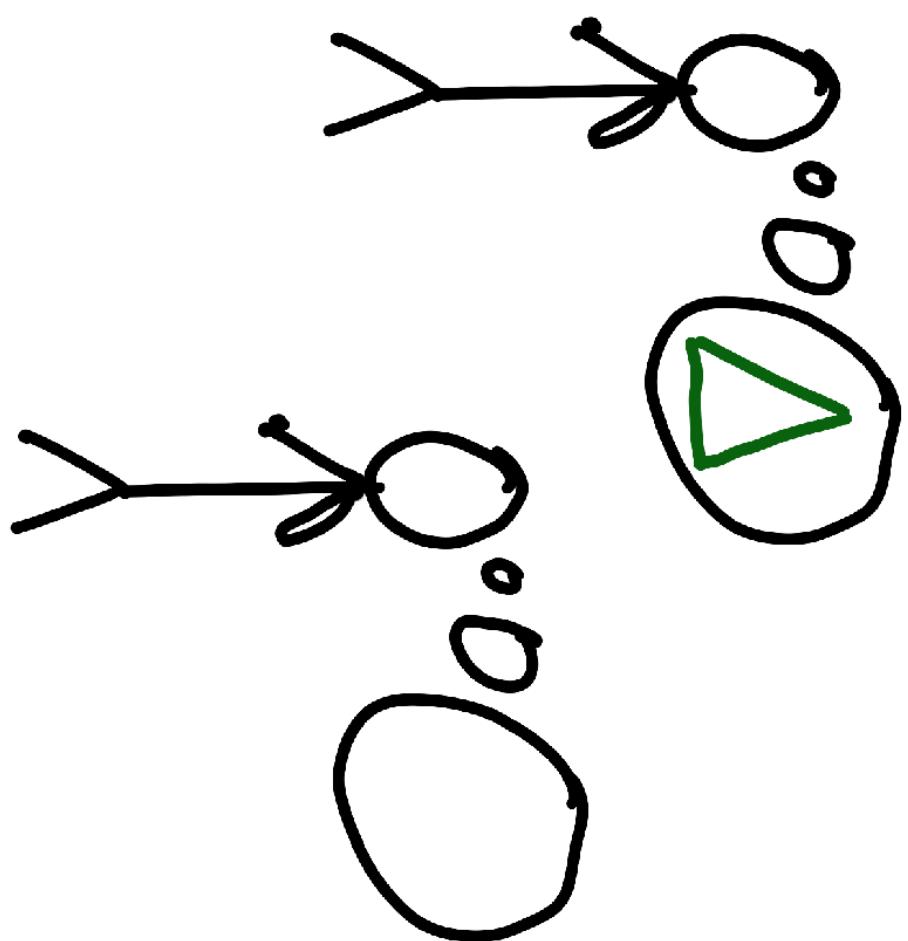
Found all consumers for
this service

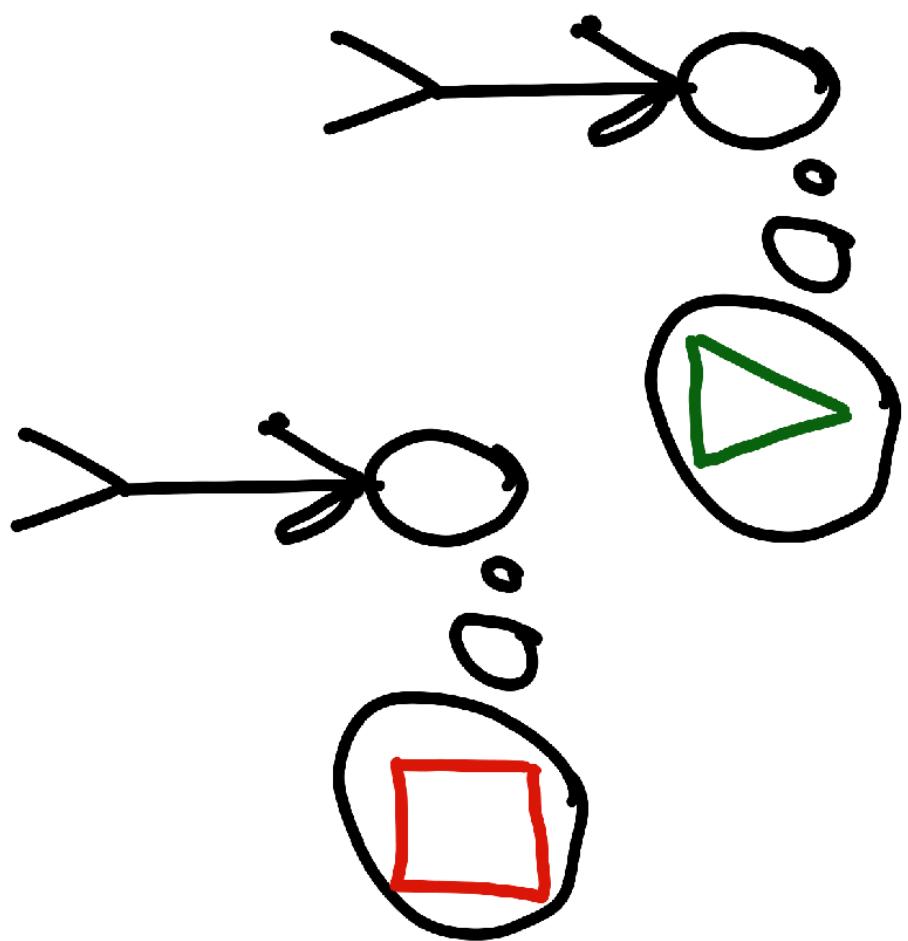
Understand what
they want

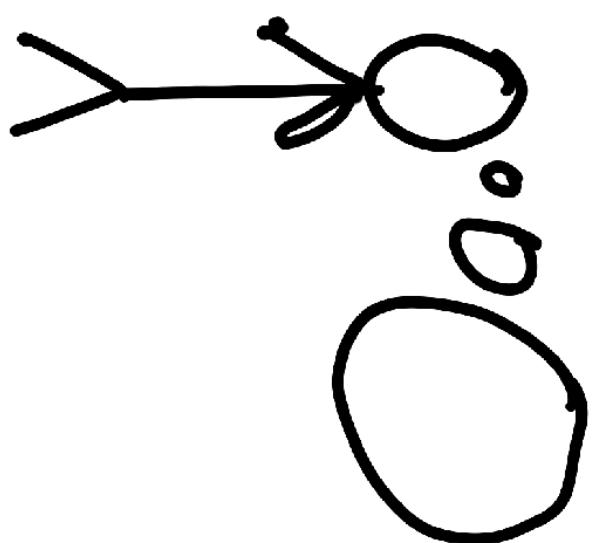
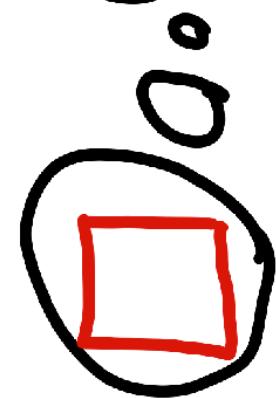
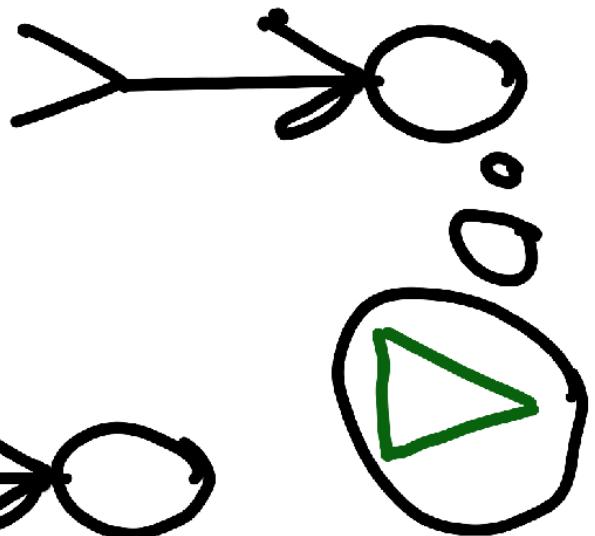
And make requirements

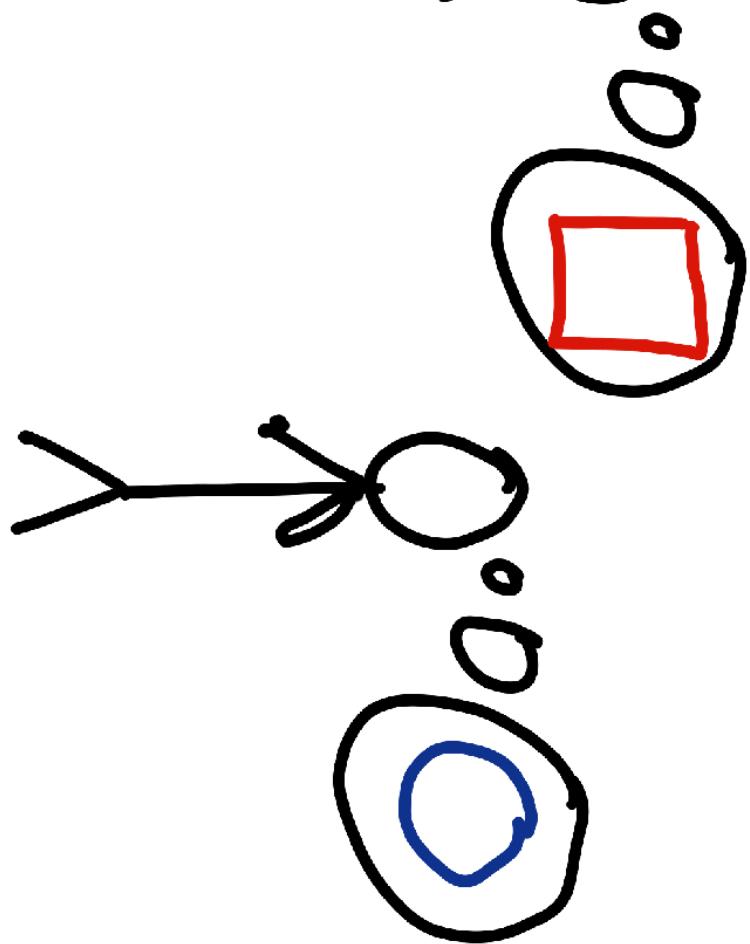
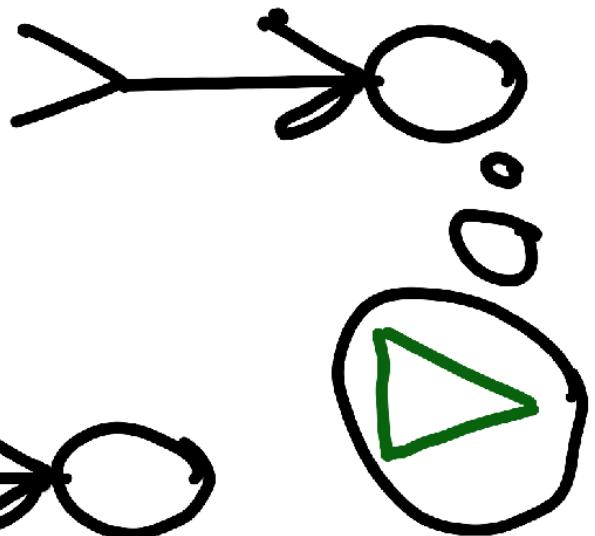


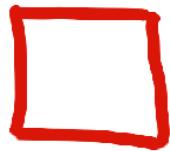
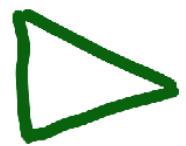


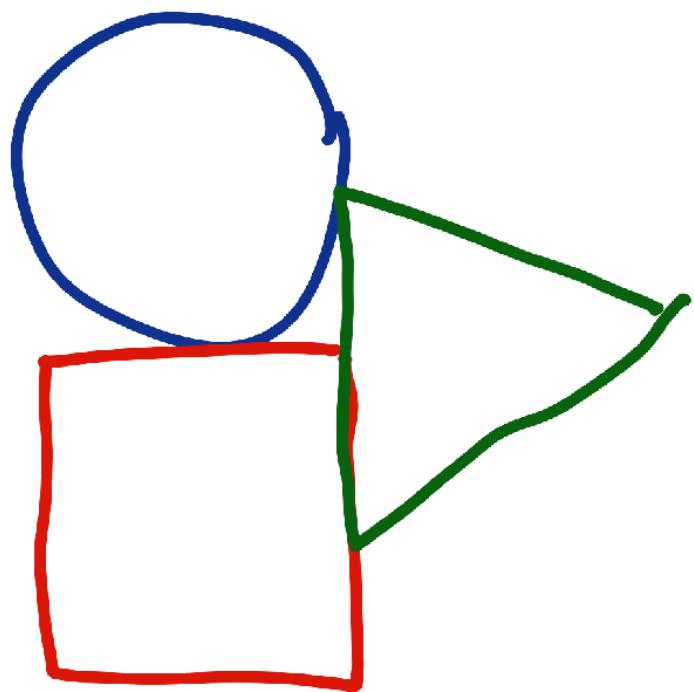


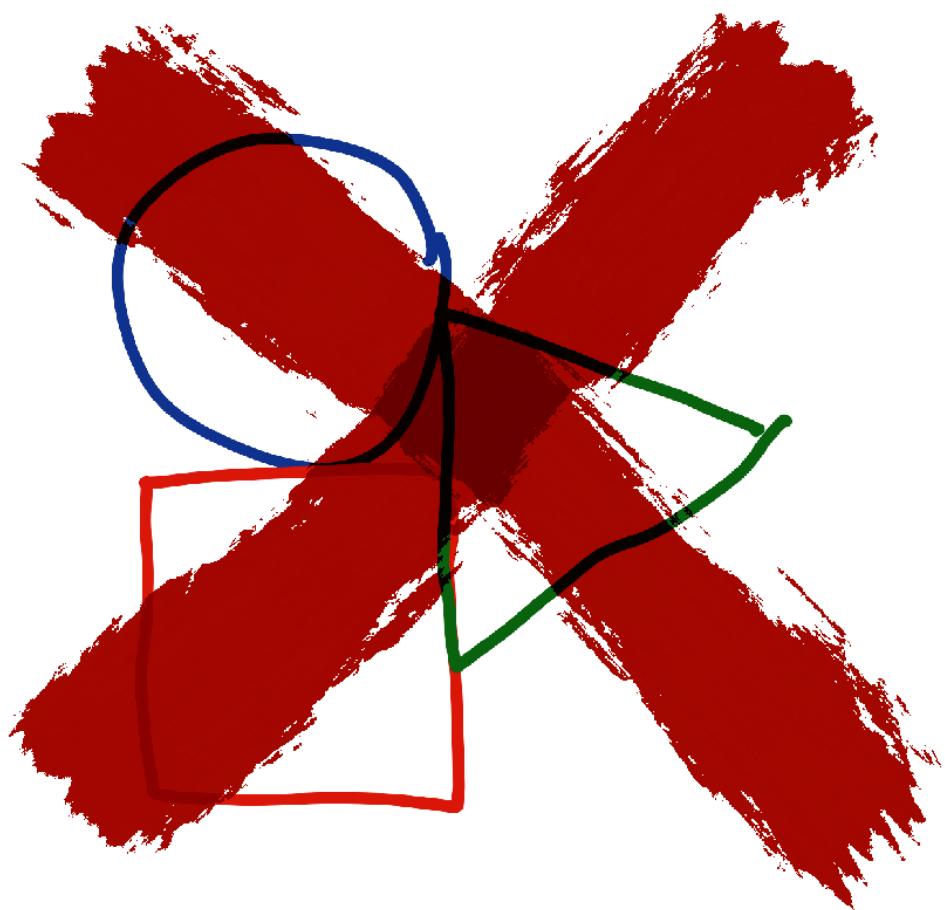


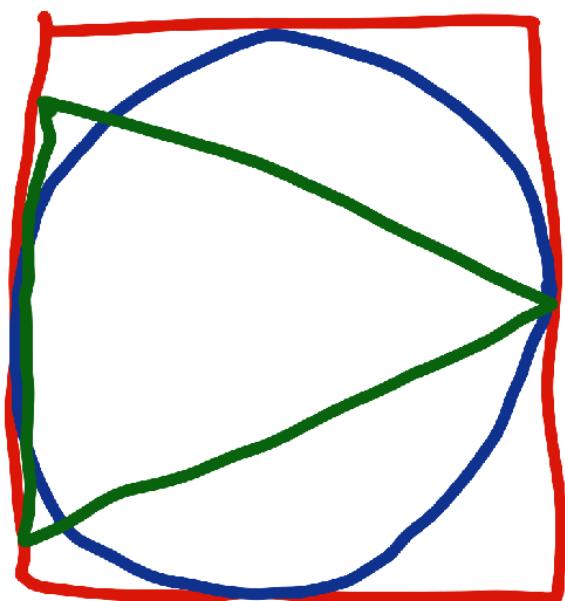


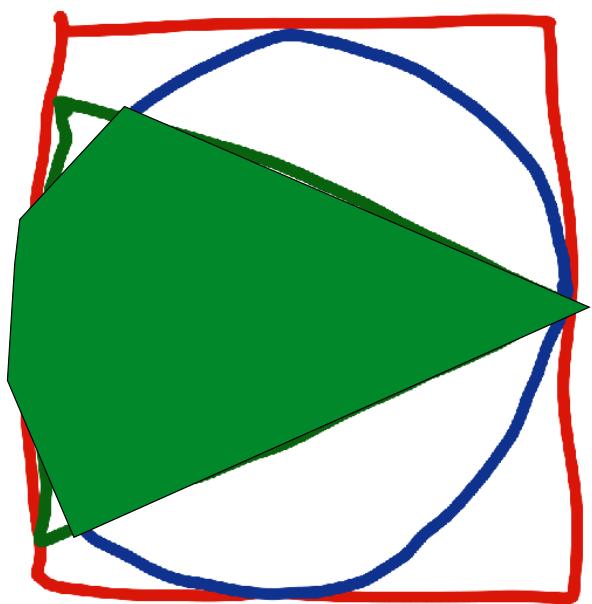


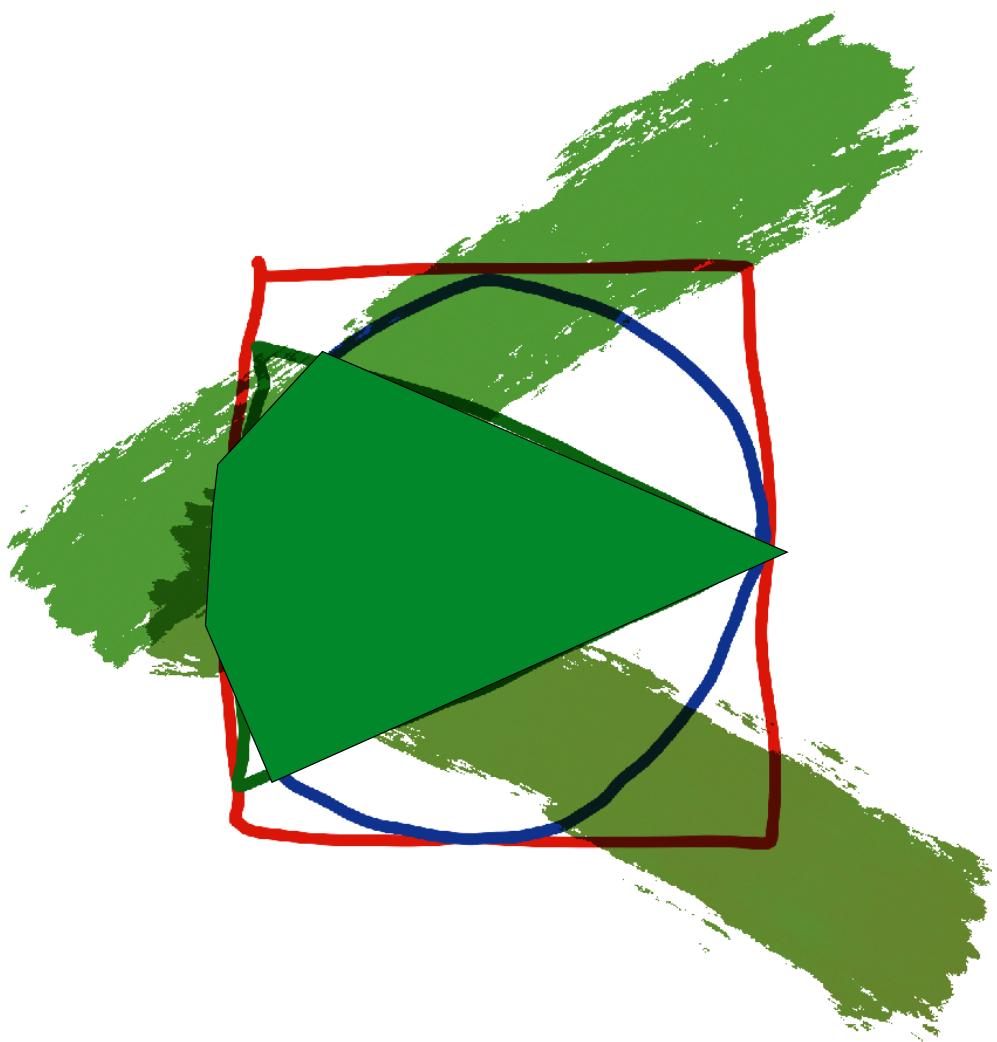












What I got

b2b

Different data for different
consumers

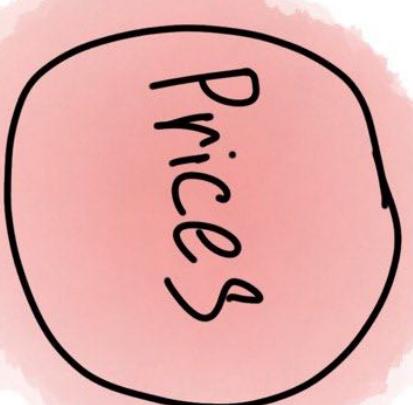
Isolated release cycle



WEB



PHONE



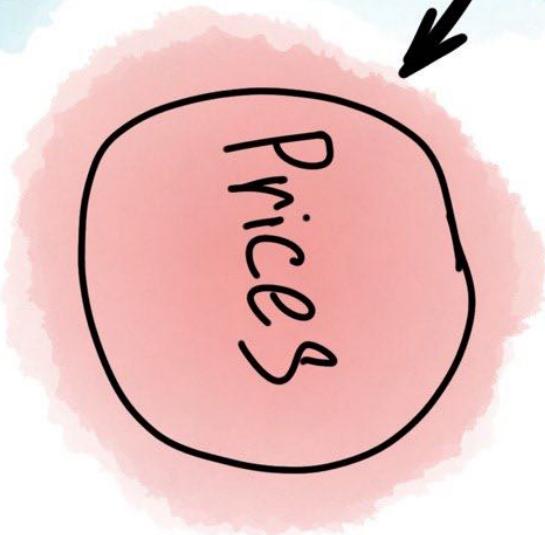
Prices

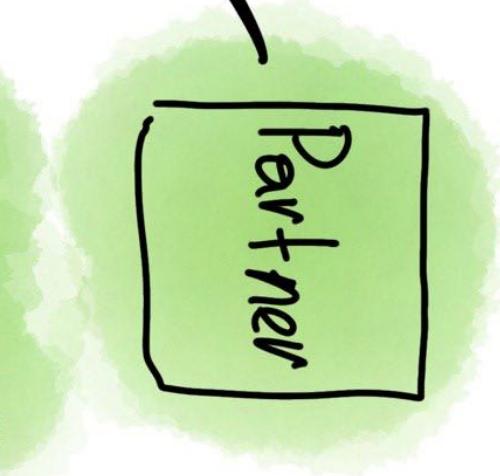
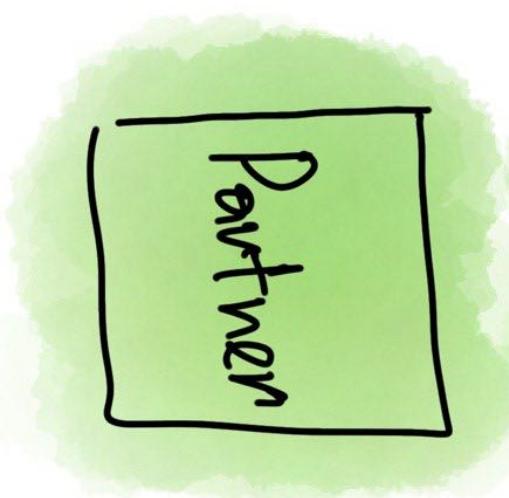
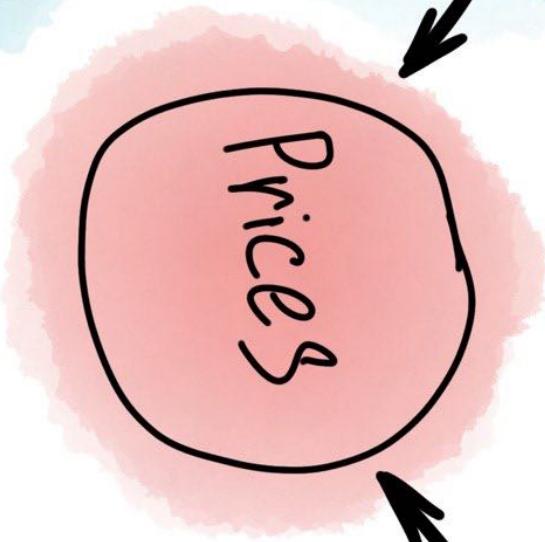


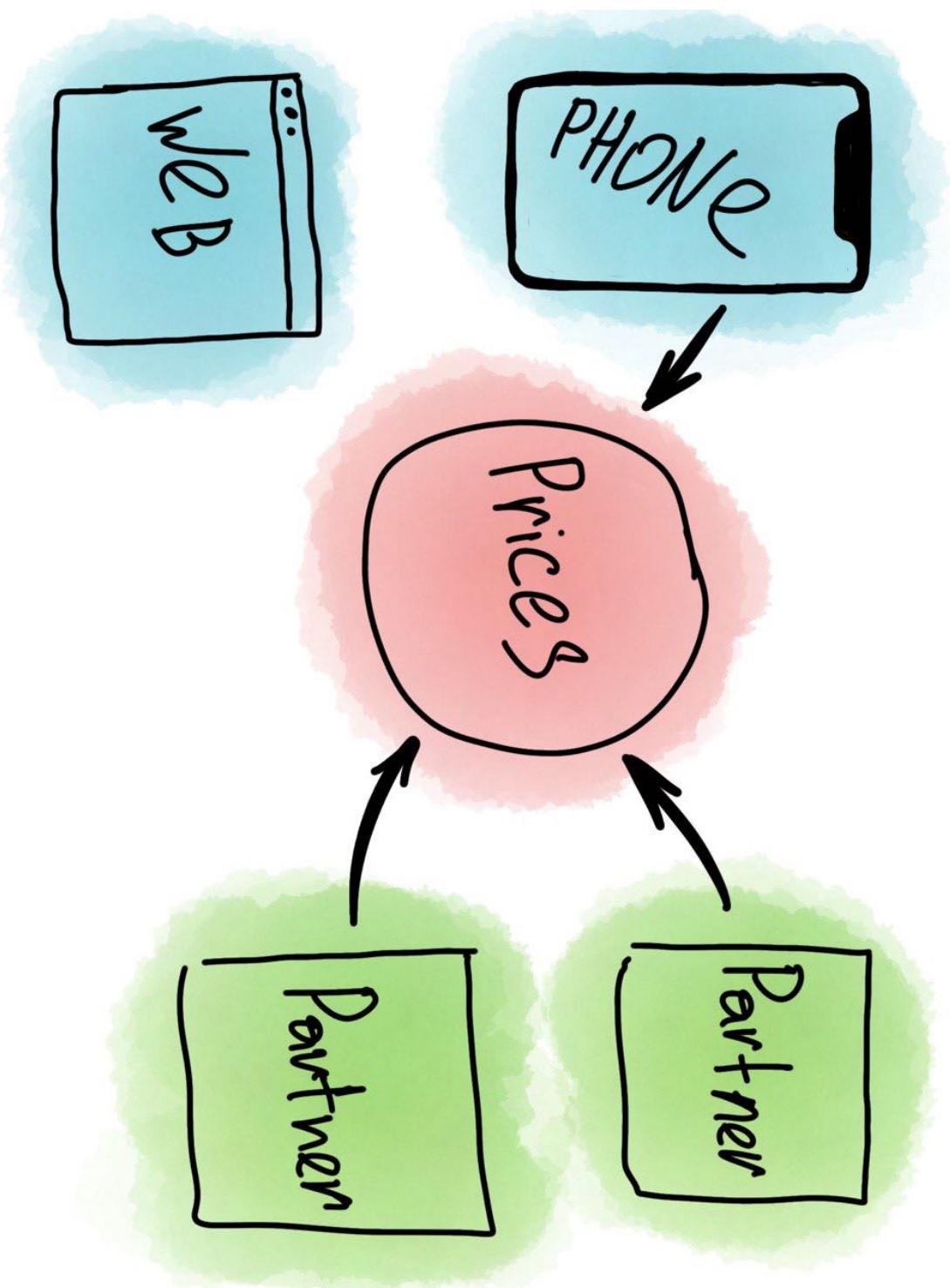
Partner

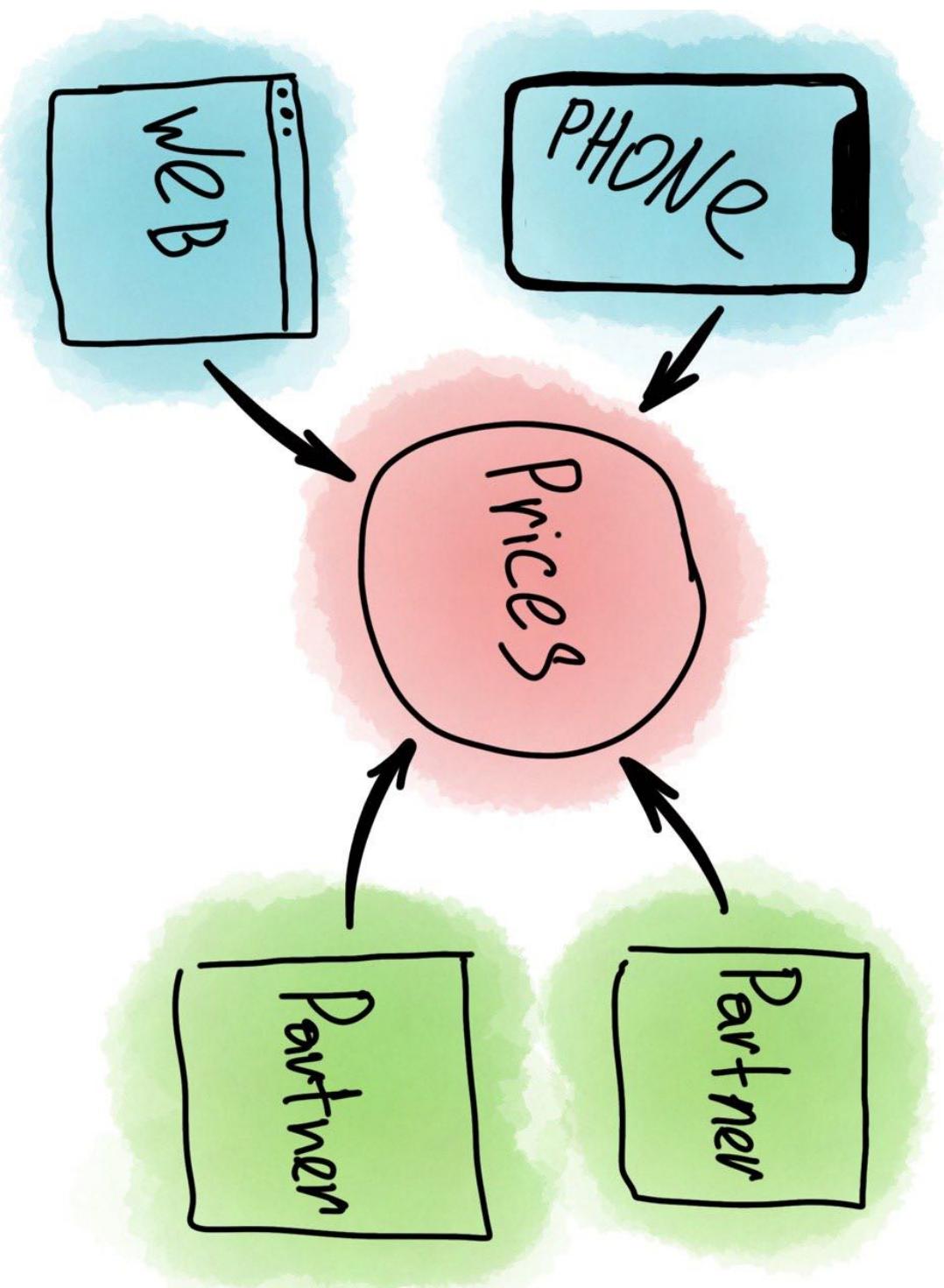


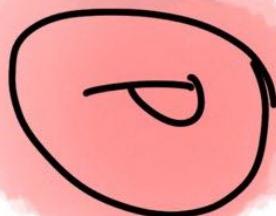
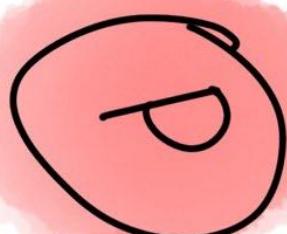
Partner

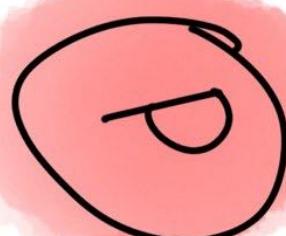


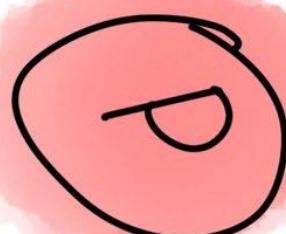


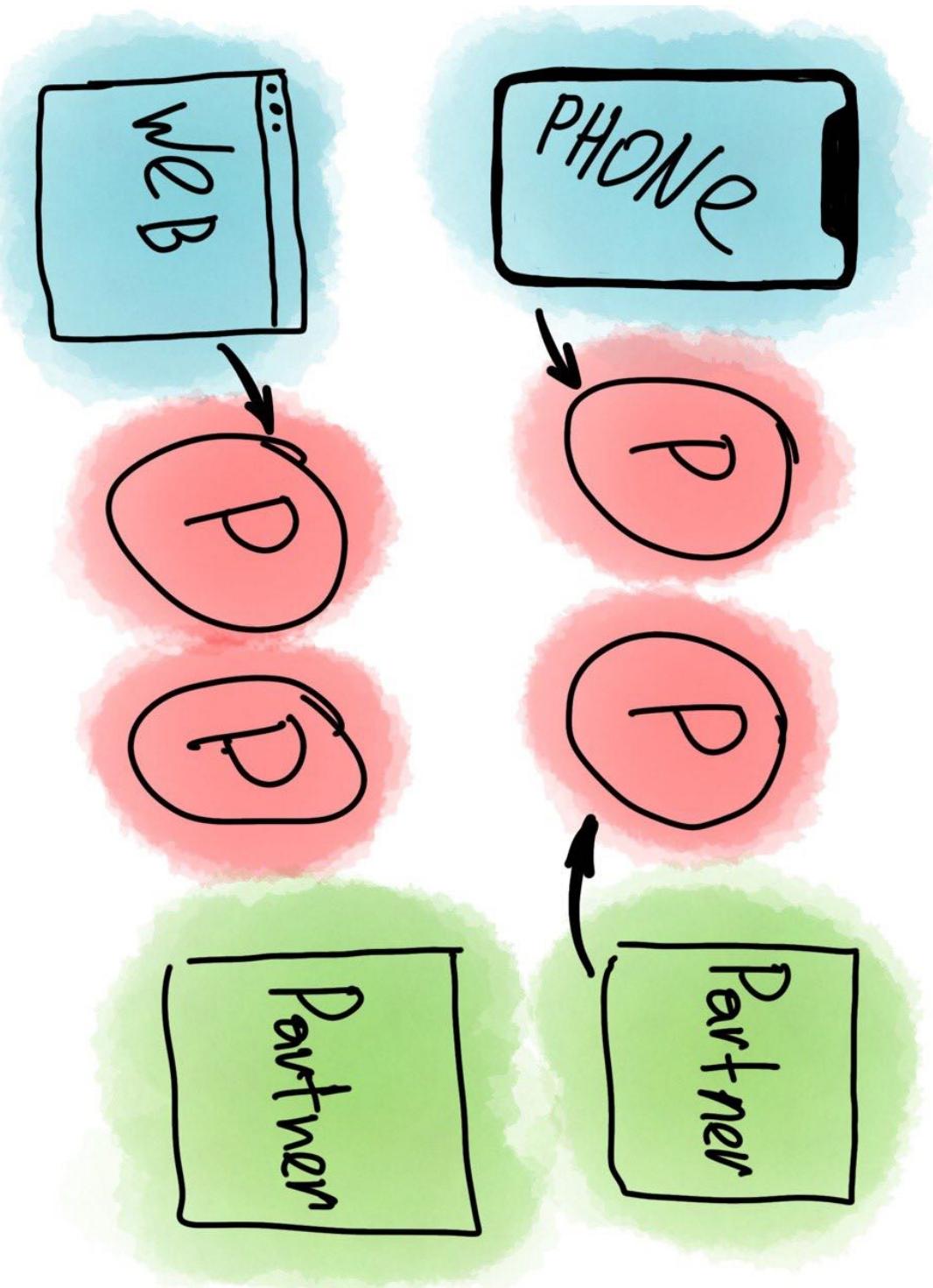


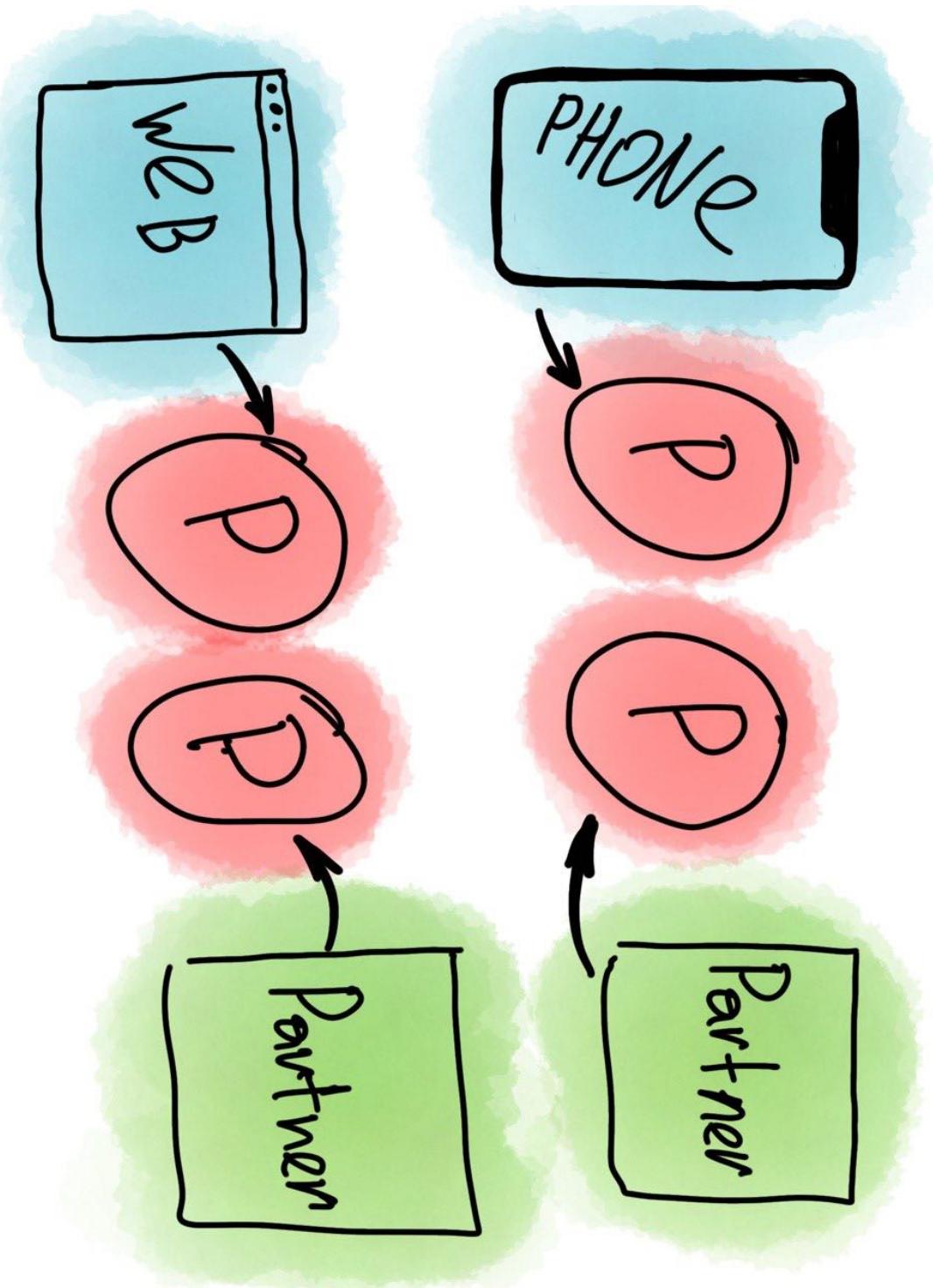












Approaches and technologies

Use best solution
for the problem



johnfound 2 января 2017 в 20:52

Писать веб-сайты на ассемблере полезно и приятно

Разработка веб-сайтов, Assembler

Из песочницы

AsmBB demo

search

asm **asmblb**

fossil qannedev heap list memfs seccore

Thread list Answer

Registration and submission



SergeyShishkin

#15155 Last edited: 29.09.2016

Hello!

And thanks to John f development. About mathematician) and c project. Ready to tri directions.

Thanks again John.

Вообще-то, это утверждение было лишь гипотезой. Чтобы доказать или отвергнуть ее, этой весной я занялся писать форум на ассемблере.

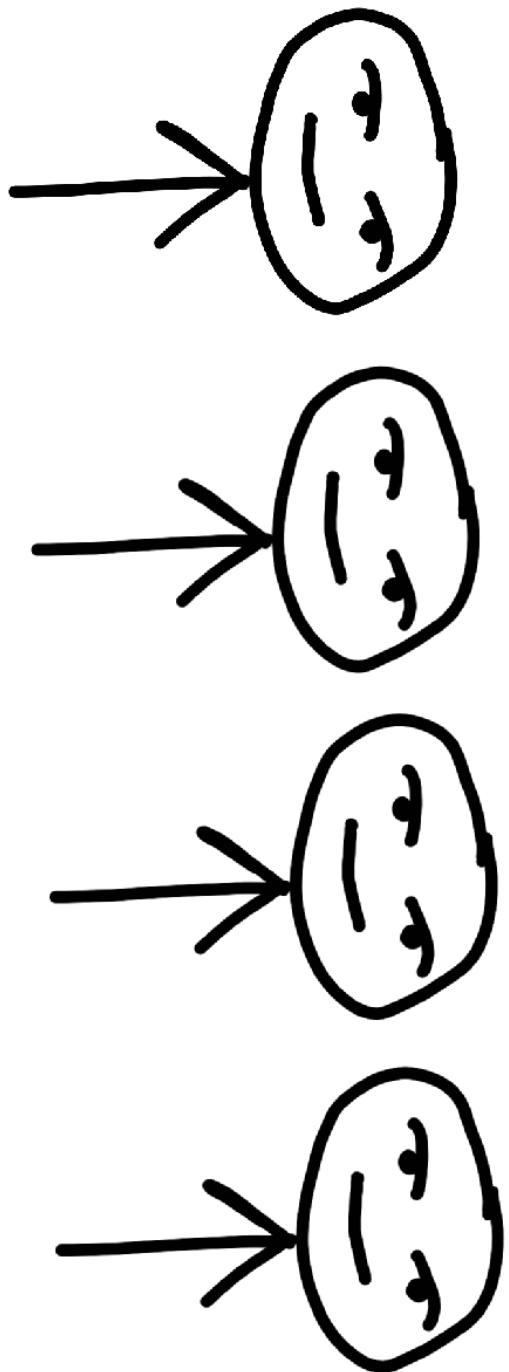
Раньше у меня уже было веб-приложение на ассемблере — CMS для малого сайта.

Только оно работает в режиме "один пишет, многие читают". При том, использует CGI интерфейс и поэтому "многие" читать одновременно тоже не получается.

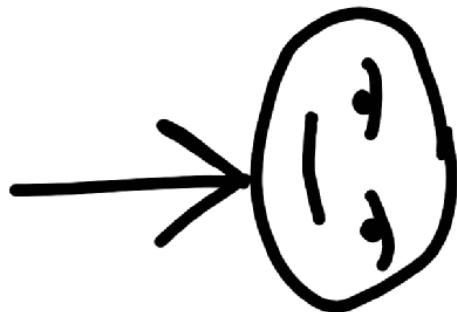
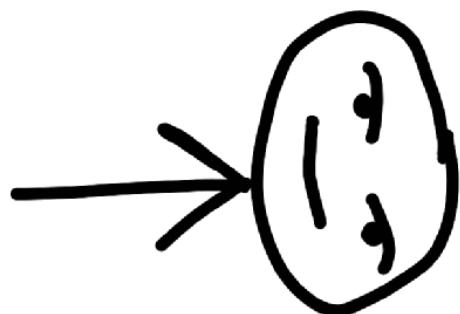
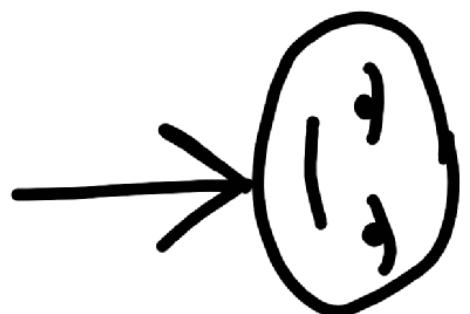
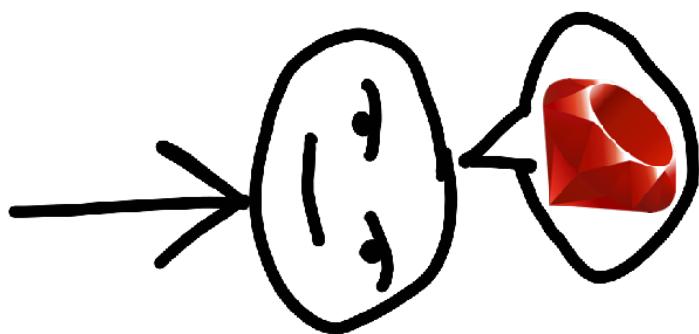


Remember about
business and team

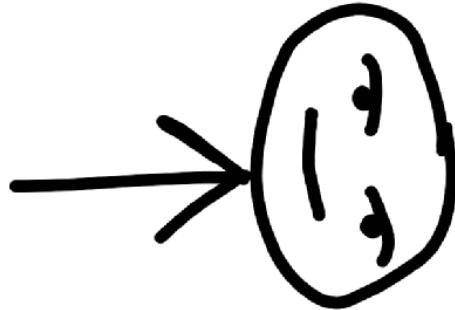
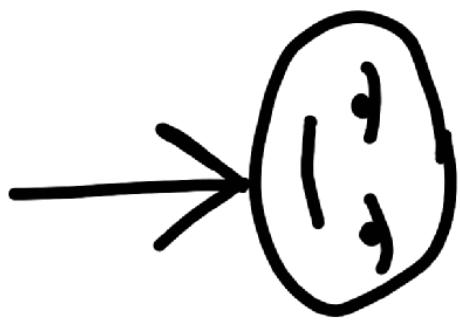
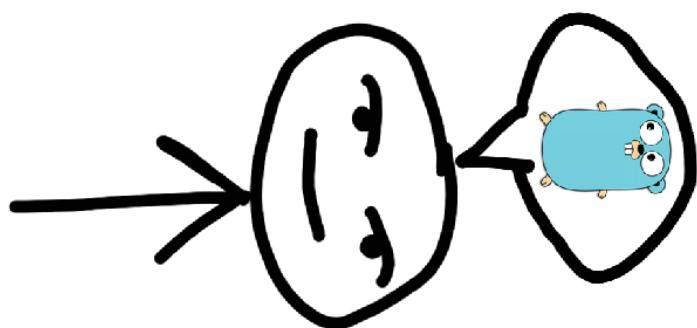
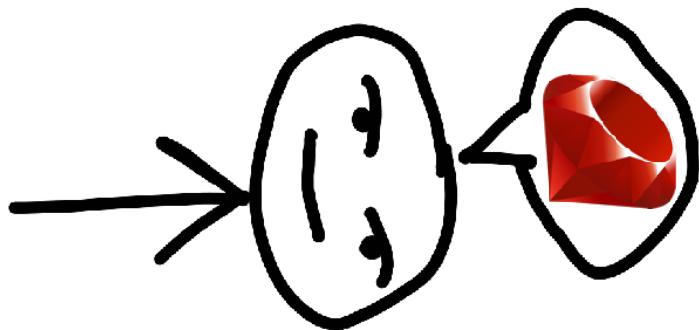
(c)



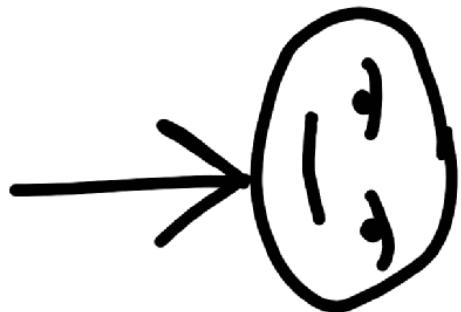
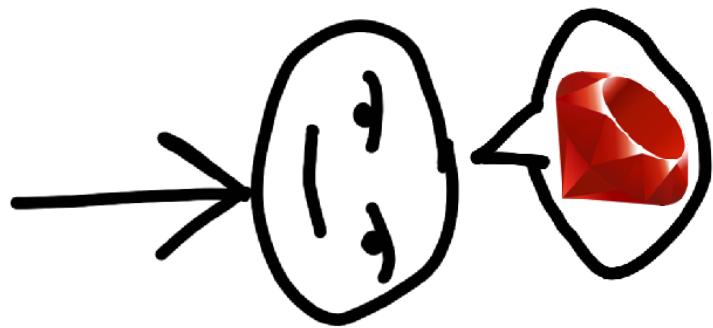
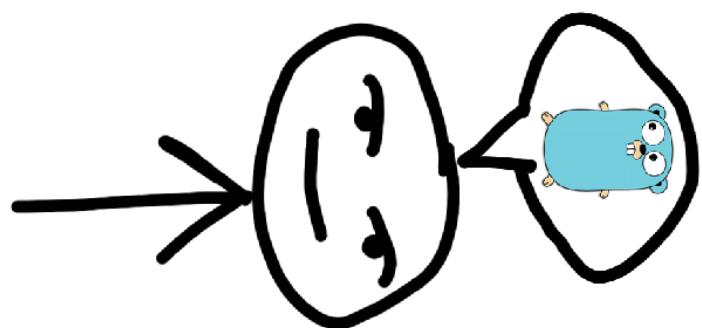
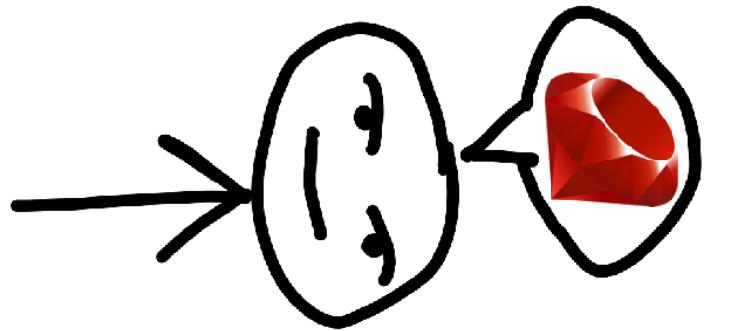
(c)



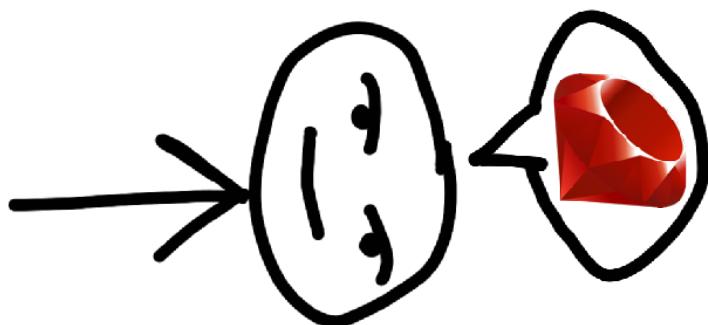
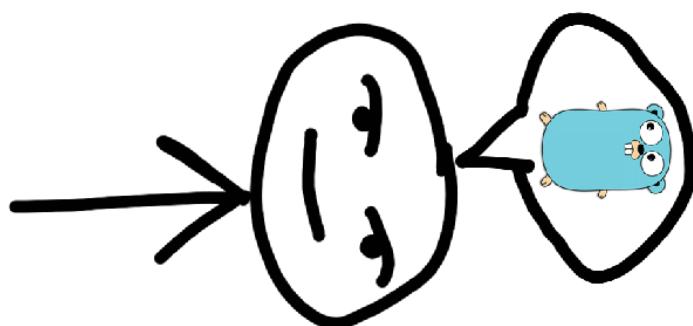
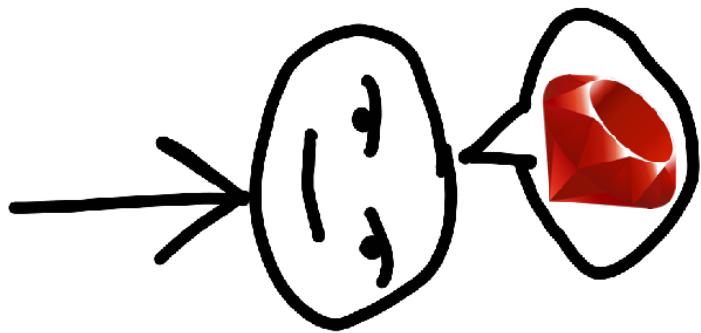
(c)
d
o
r



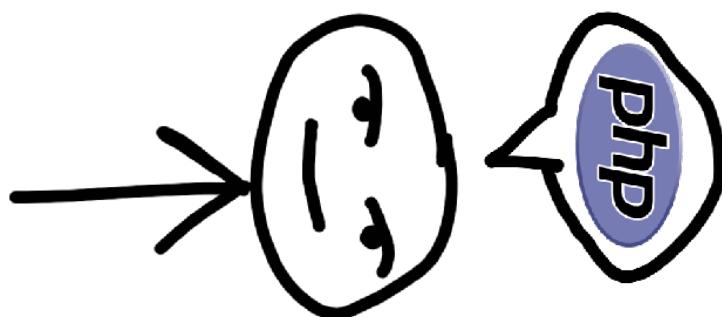
(C)
d
o
g

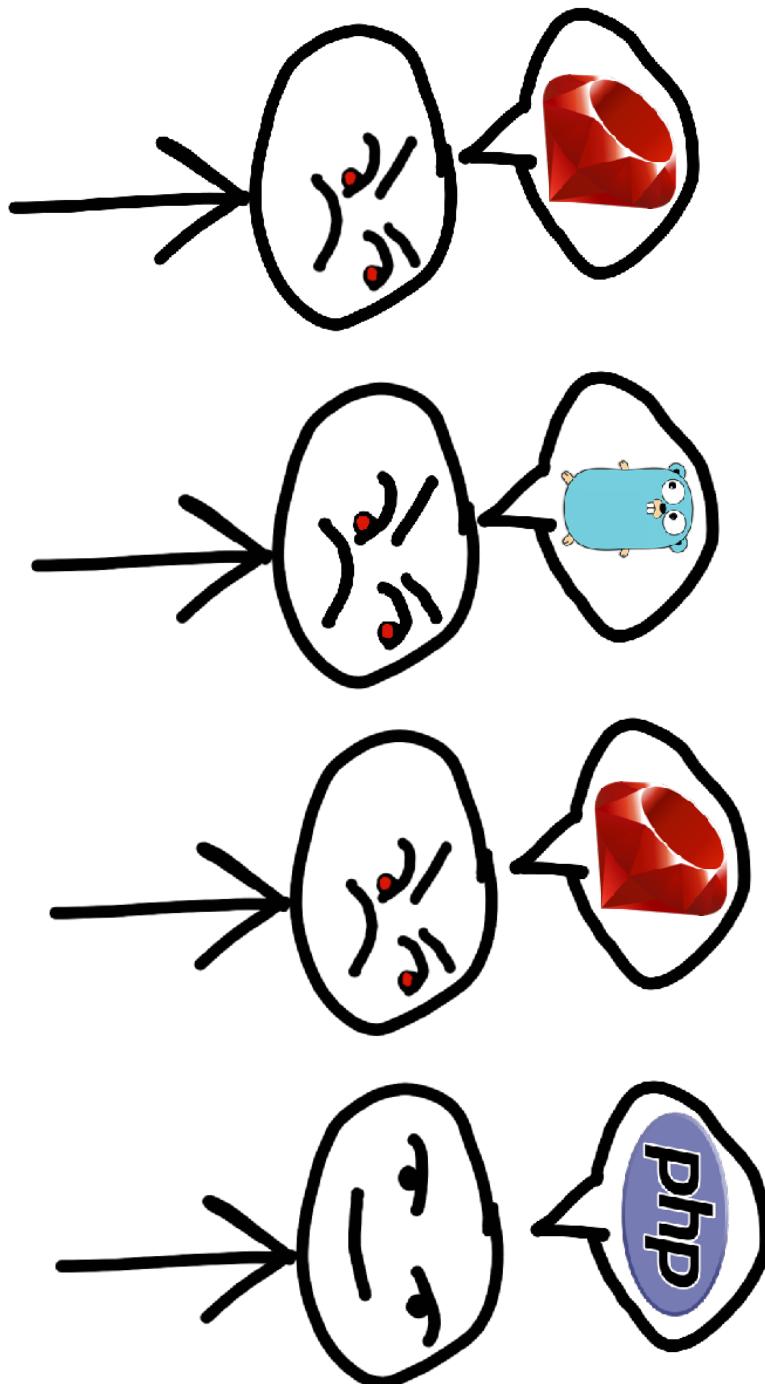


۱۰۷

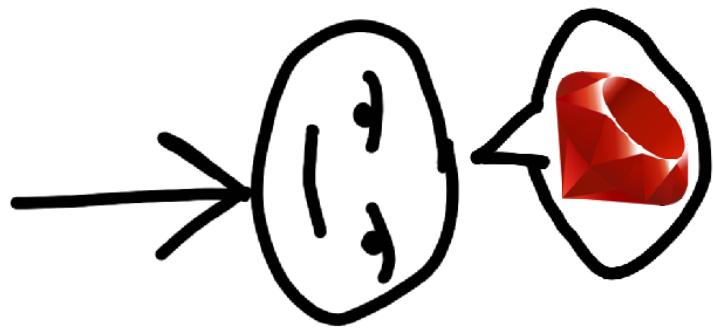
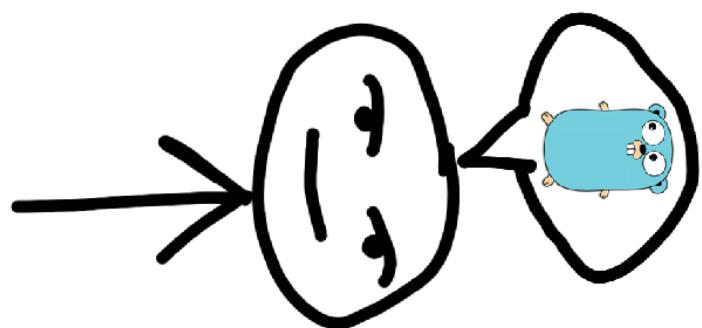
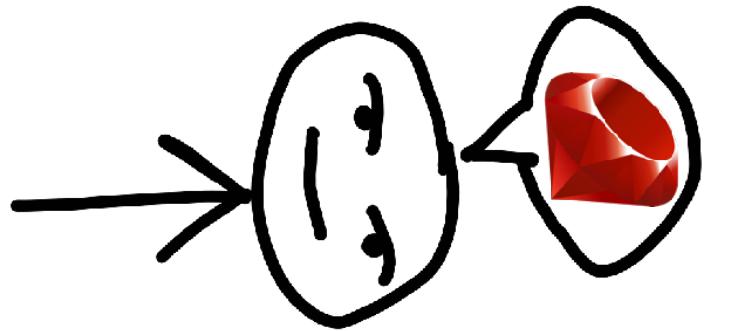


;) ;)





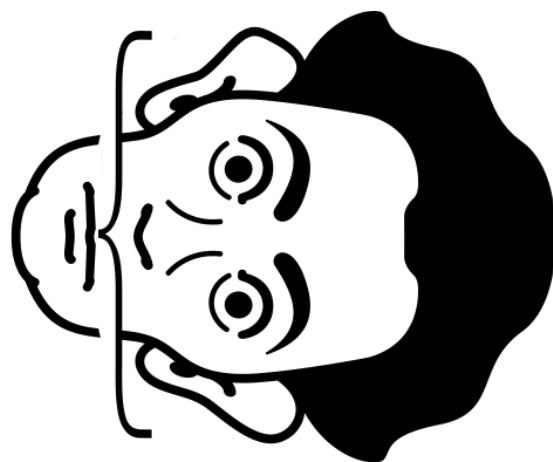
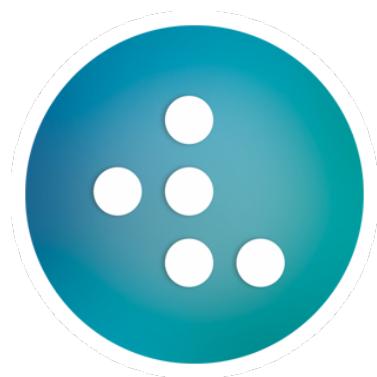
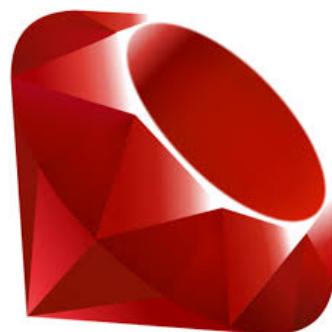
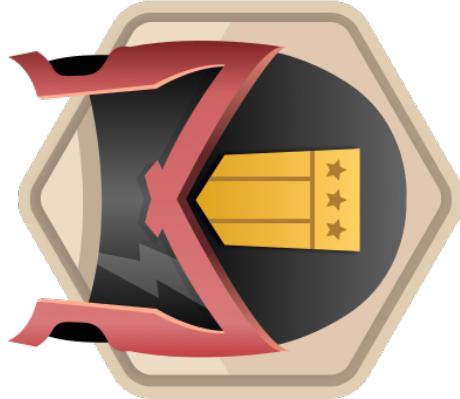
α
|
 α



卷之三

Suicide Squad



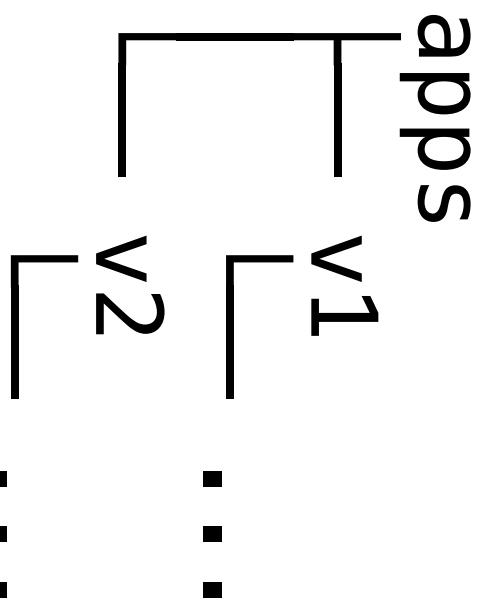


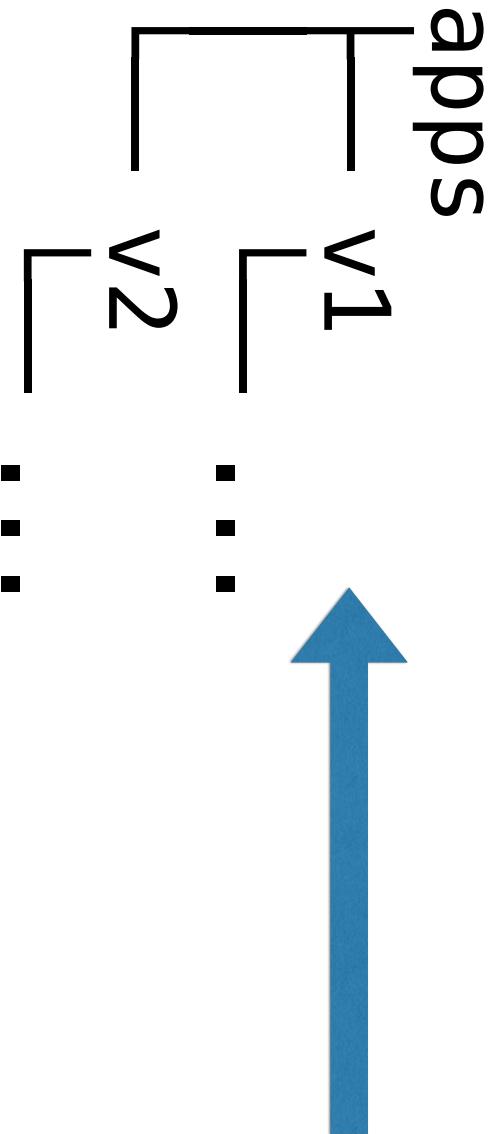
hanami

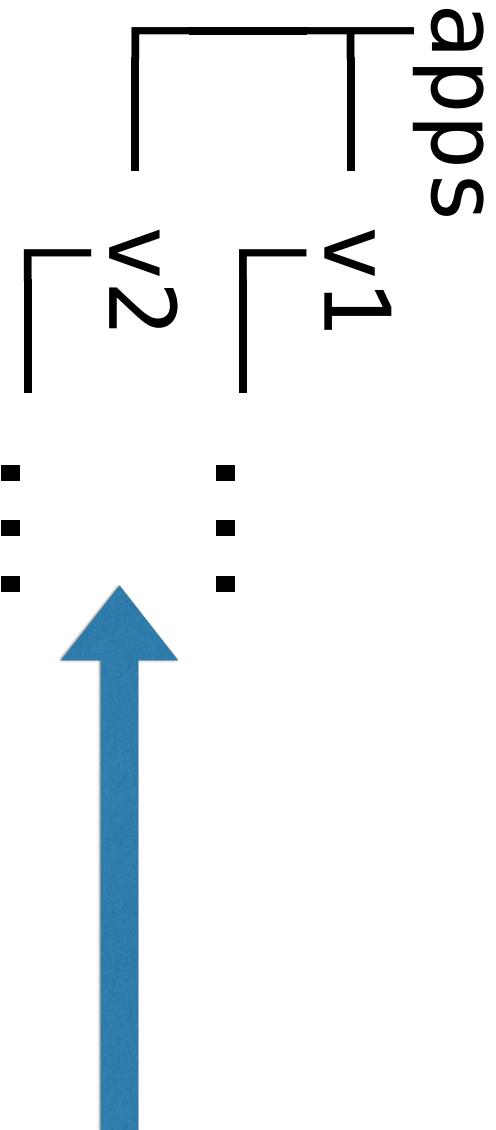
Why?

Because I can

One app, one version







```
V1::Controllers::Prices::Index.new  
V1::Controllers::PartnerPrices::Index.new
```

dry-system

Why?

- o Easy to work with dependencies
- o Easy for testing (DI)
- o Loading without any magic

t.me/pepeprogramming

Presenters

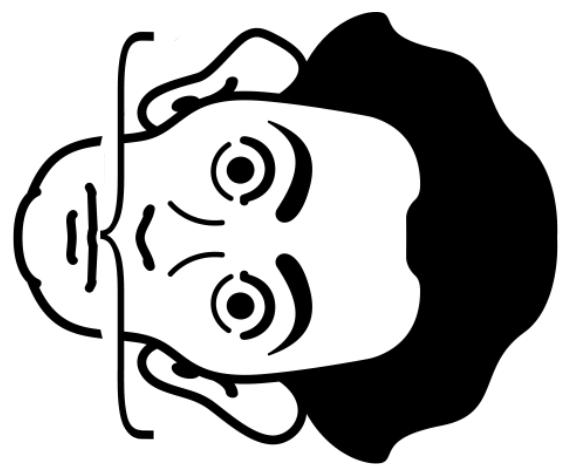
Related to action instead
model

github.com/safelylaunch/core

```
v1
  |
  +-- controllers
    +-- partners_prices
      +-- index.rb
      +-- prices
        +-- index.rb
    +-- presenters
      +-- entities
        +-- price.rb
      +-- prices
        +-- index
          +-- error.rb
          +-- success.rb
    +-- partners_prices
      +-- index
        +-- error.rb
        +-- success.rb
```

```
v1
  └── controllers
      └── partners_prices
          └── index.rb
              ├── prices
              └── index.rb
      └── presenters
          ├── entities
          │   └── price.rb
          └── prices
              └── index
                  ├── error.rb
                  └── success.rb
  └── partners_prices
      └── index
          ├── error.rb
          └── success.rb
```

```
v1
  controllers
    partners_prices
      index.rb
    prices
      index.rb
  presenters
    entities
      price.rb
    prices
      index
        error.rb
        success.rb
  partners_prices
    index
      error.rb
      success.rb
```



Surrealist

```
module Api
  module Presenters
    module V1
      module Check
        class Success
          include Surrealist

          json_schema do
            { key: String, enable: Bool }
          end

          attr_reader :key, :enable

          def initialize(key:, enable:)
            @key = key
            @enable = enable
          end
        end
      end
    end
  end
end
```

```
v1
└── controllers
    └── partners_prices
        └── index.rb
        └── prices
            └── index.rb
└── presenters
    └── entities
        └── price.rb
└── prices
    └── index
        └── error.rb
        └── success.rb
└── partners_prices
    └── index
        └── error.rb
        └── success.rb
```

Full structure of
the service

```
apiary.apib
dredd.yml
apps
  v2
  v1
lib
core
  entities
  libs
  repositories
  types
  internal
  operations
  partners
  operations
  types.rb
scripts
  dredd_server.sh
spec
  contracts
    v1_hook.rb
system
  boot
    logger.rb
  container.rb
core_ext.rb
```



DREDD

No more outdated API Documentation

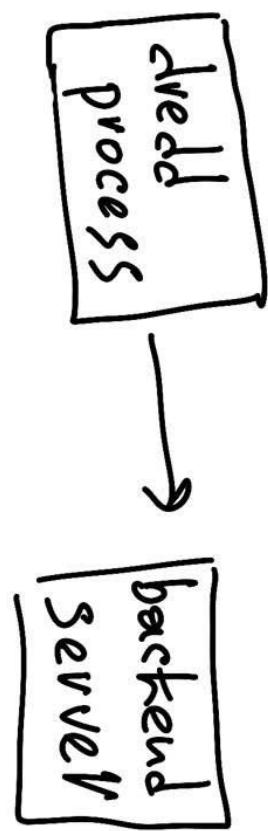


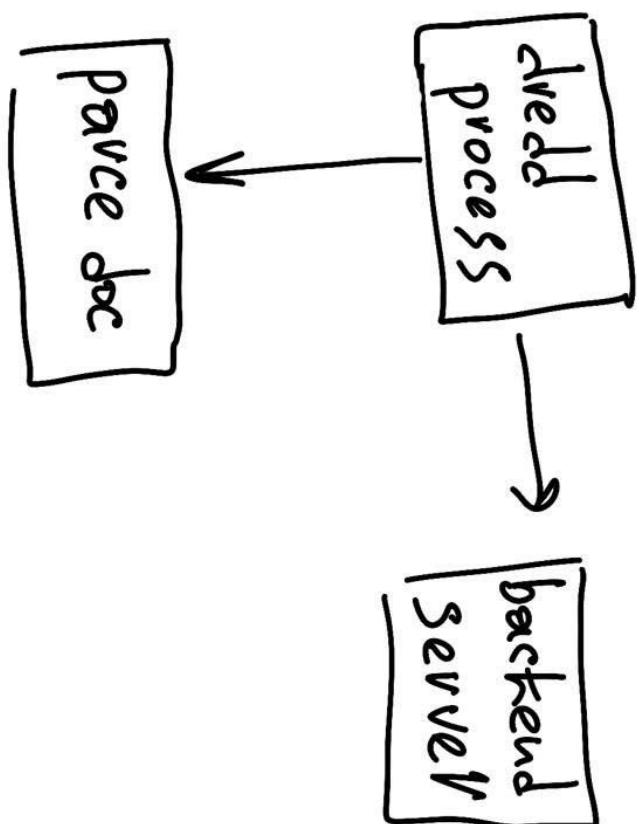
dread
process

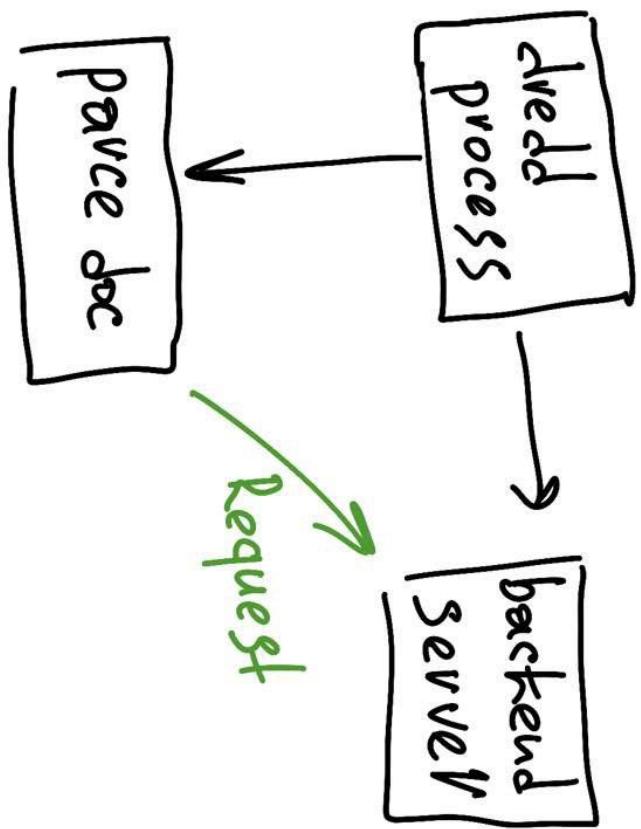
dredd
process

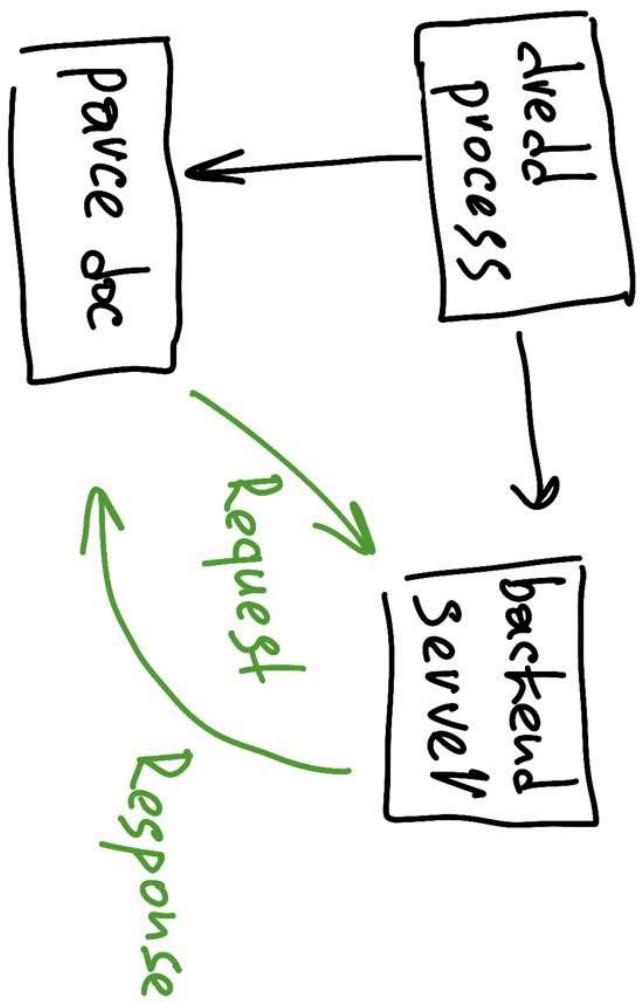
config.yaml

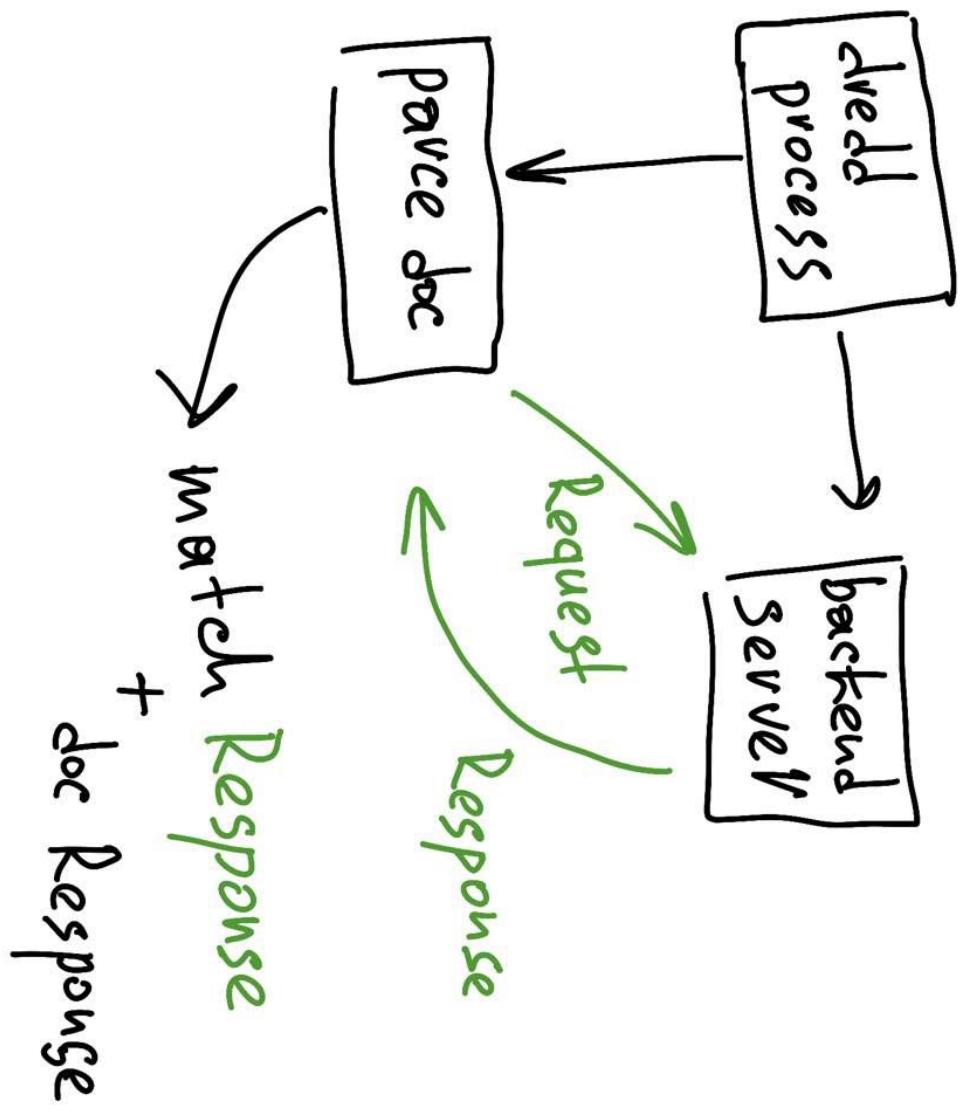
+
API documentation

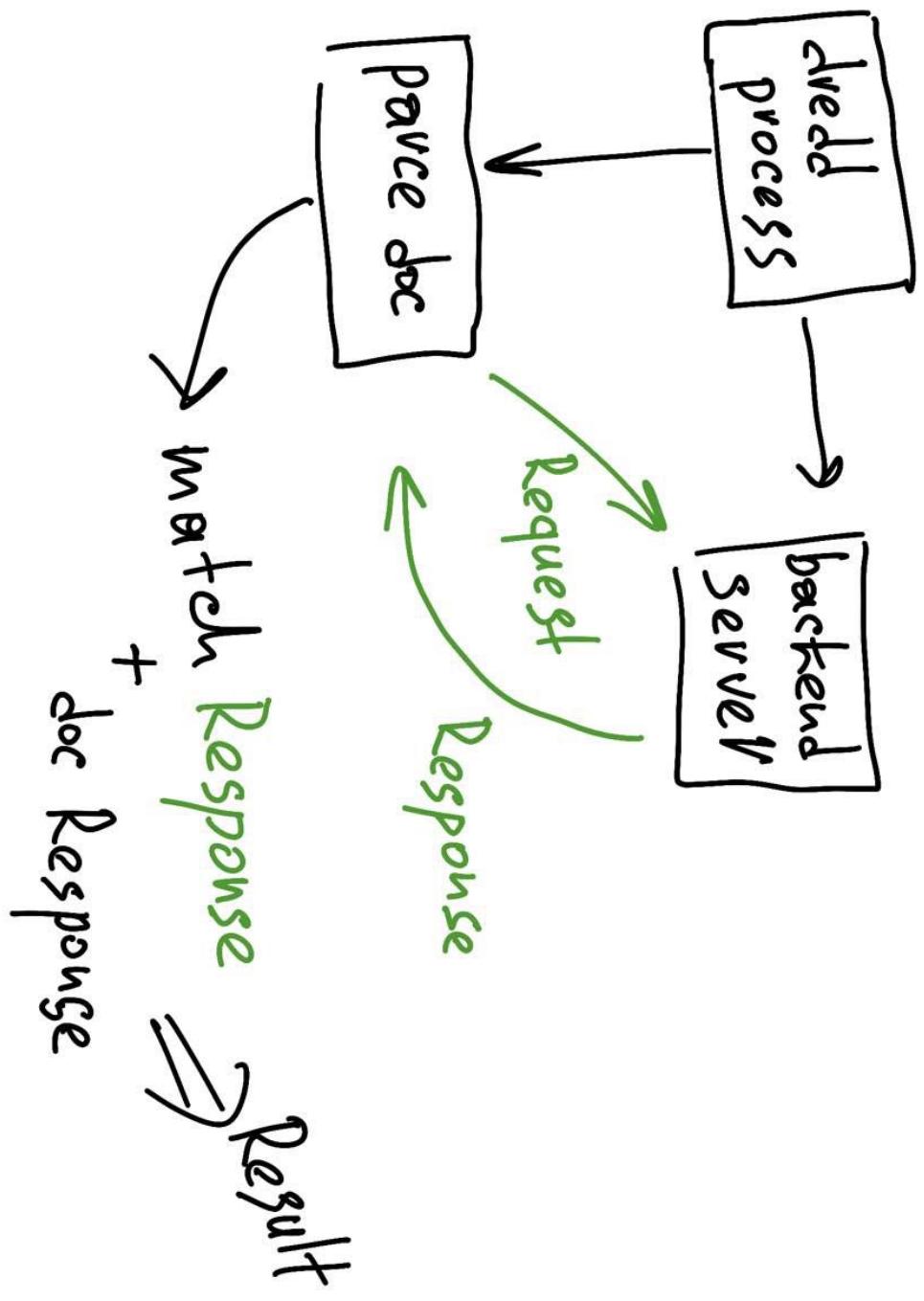












Why?

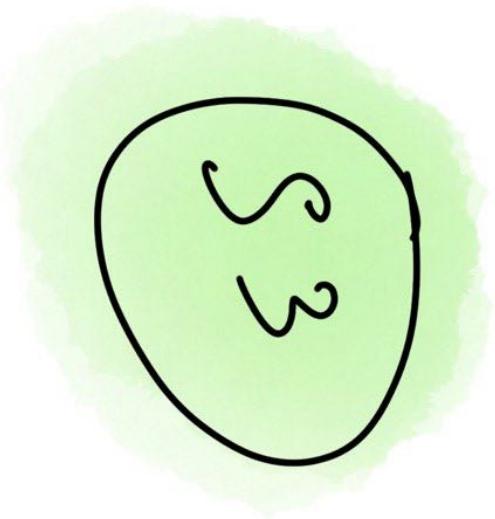
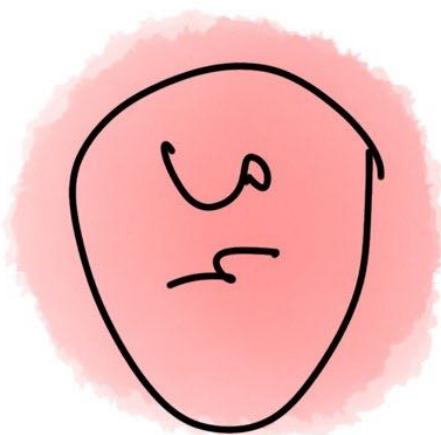
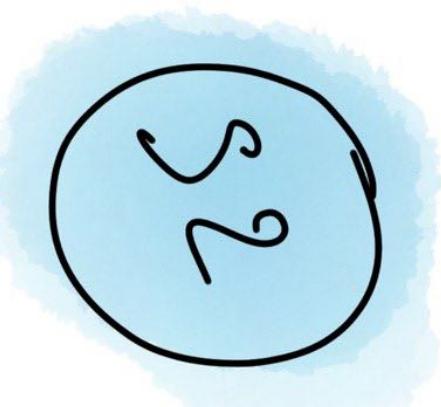
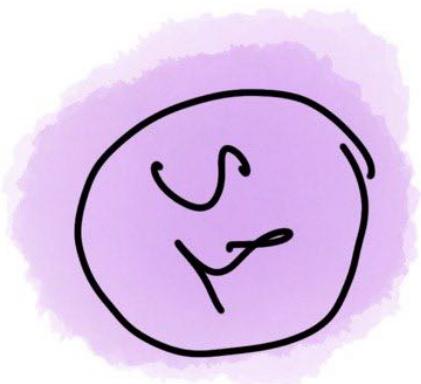
Make documentation for
JSON API

Be sure that consumers will work correctly

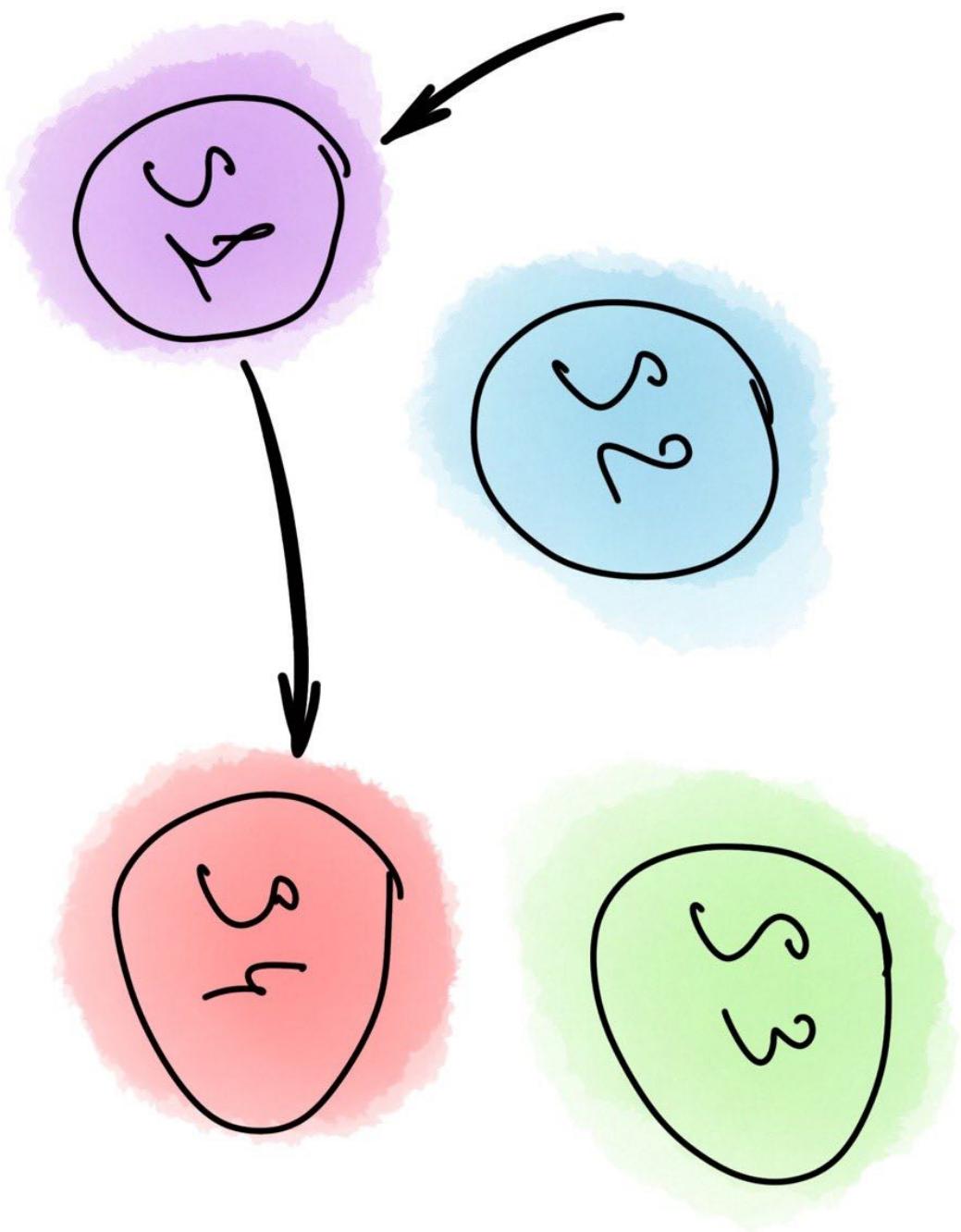
Tracing

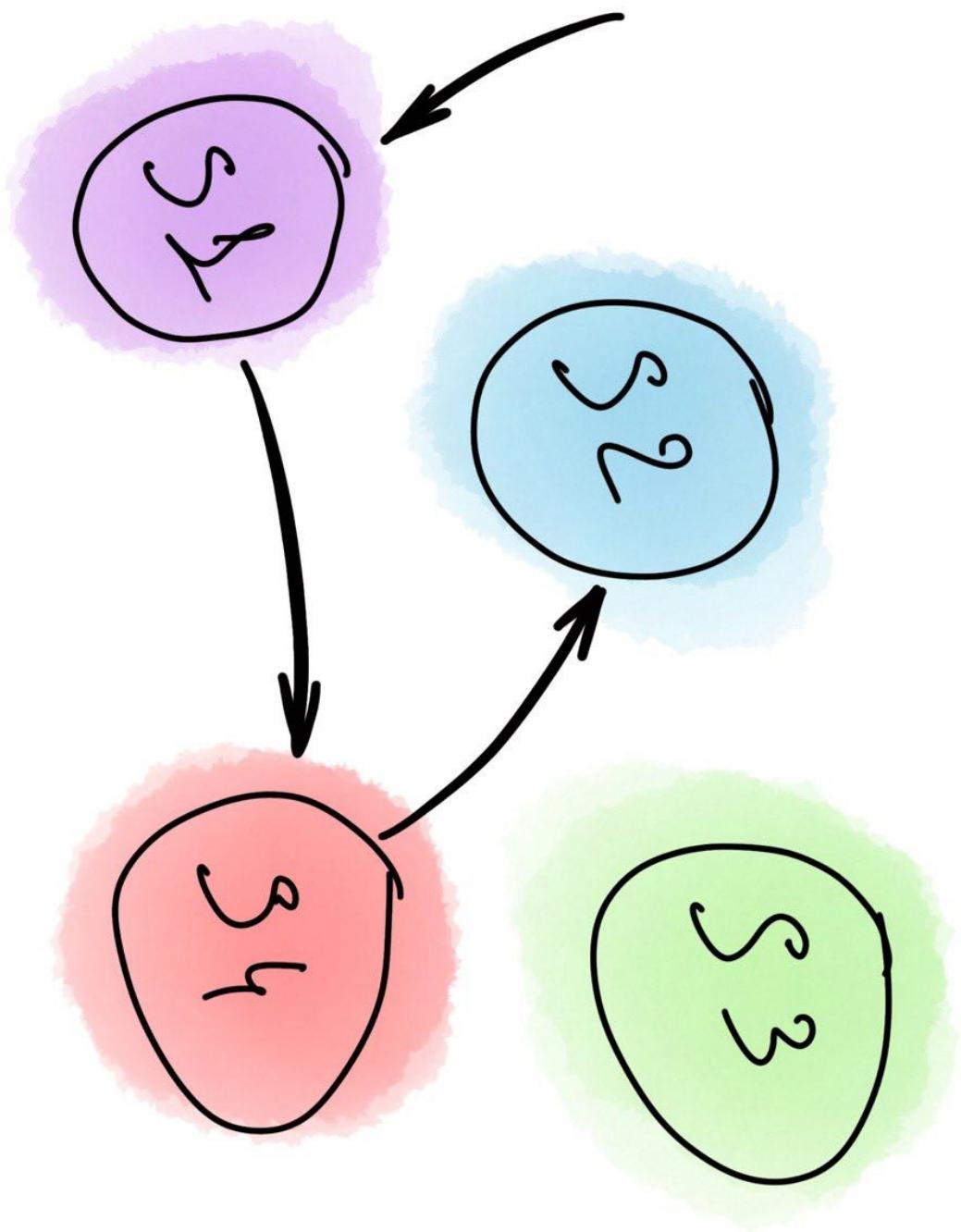
May

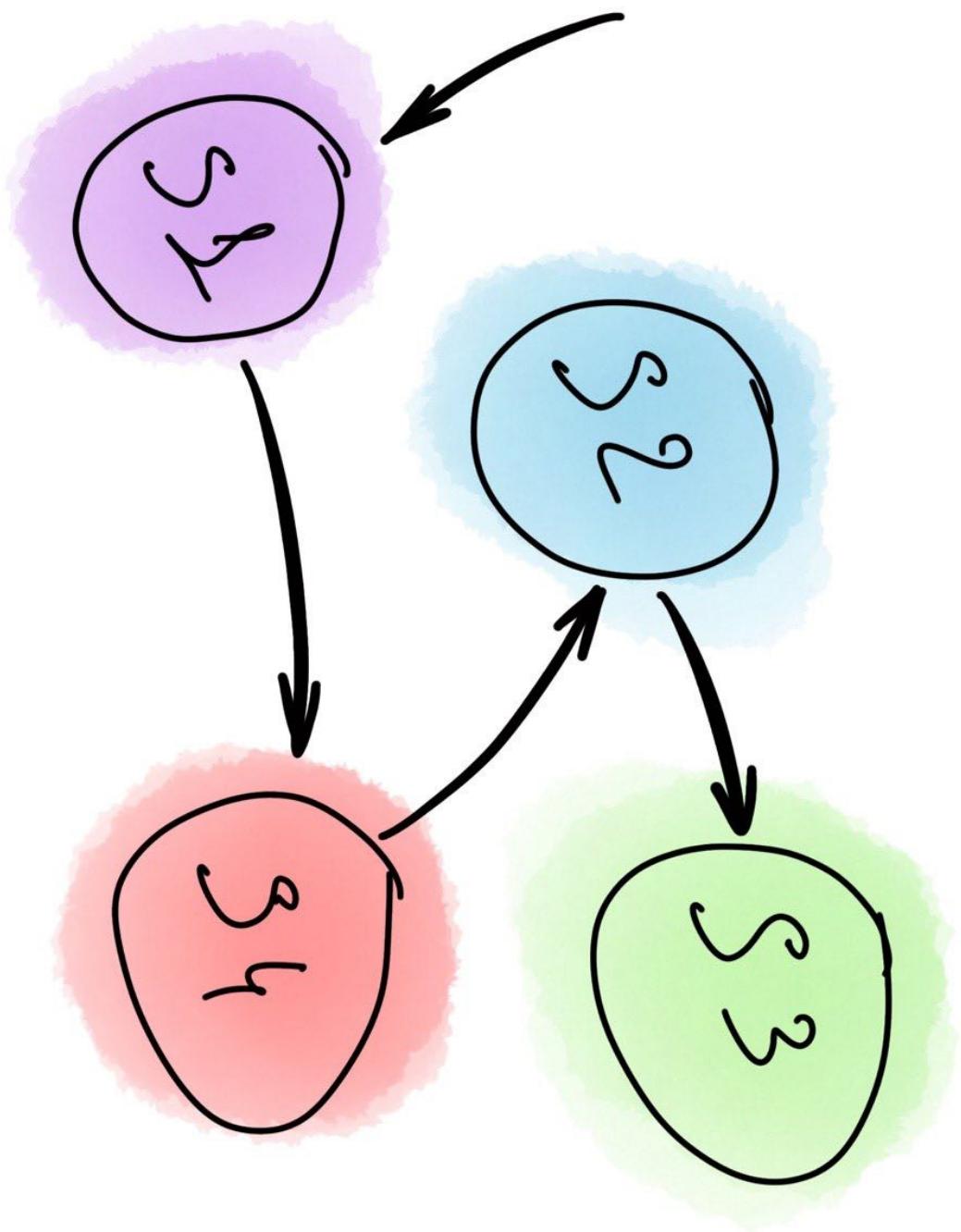
Trace full request in all services

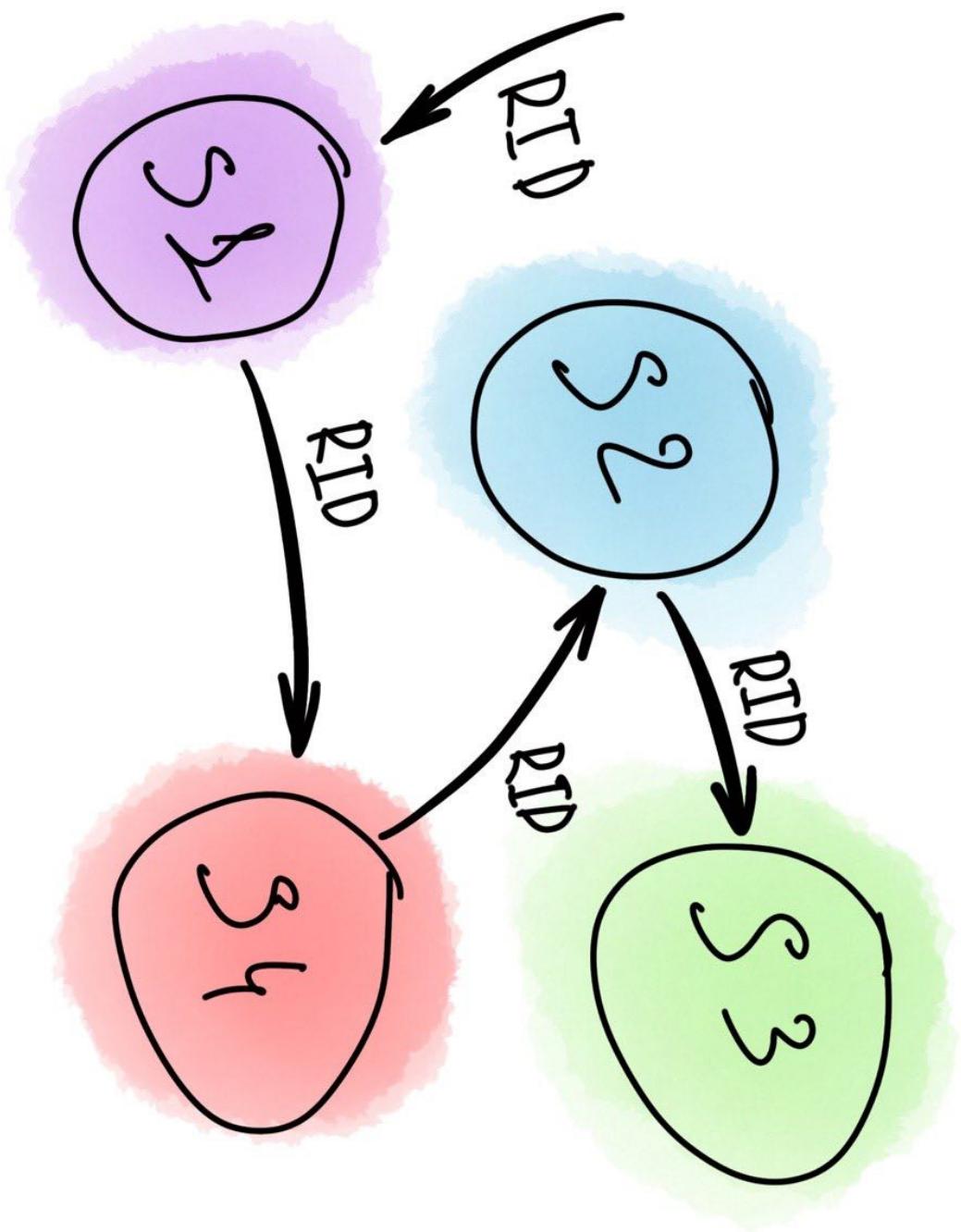












zippkin.io

semantic logger + tags =
Λ3

```

3 # Simple middleware for generating request id and tagged logger for this value
4 class RequestId
5   FORM_HASH = 'rack.request.form_hash'
6
7   # Constructor method for request id rack middleware
8   #
9   # @param [App] app rack app
10  # @param [SemanticLogger::Logger] logger logger instance
11  def initialize(app, logger = Container[:logger])
12  @app = app
13  @logger = logger
14  end
15
16  # Tag all logger calls with request id information
17  #
18  # @param [Hash] env rack env
19  def call(env)
20  env['request_id'] = SecureRandom.hex
21  env[FORM_HASH] ||= {}
22  env[FORM_HASH]['request_id'] = env['request_id']
23
24  @logger.tagged(request_id: env['request_id']) do
25    @logger.info 'test'
26
27  @app.call(env)
28  end
29  end
30  end

```

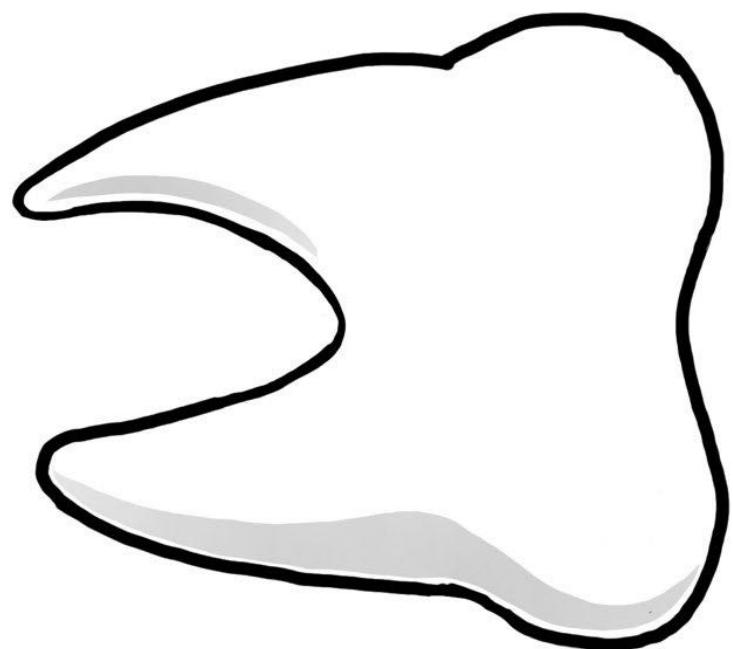
Scientist

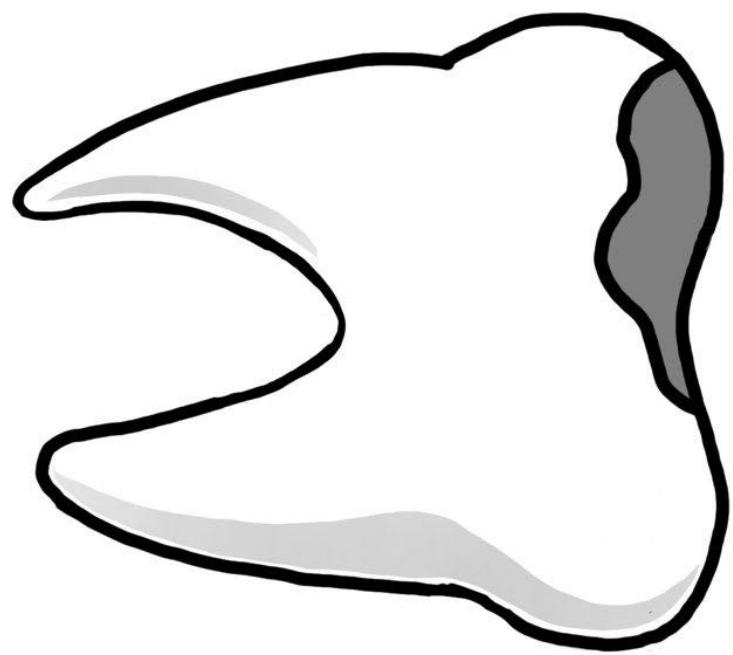
github.com/github/scientist

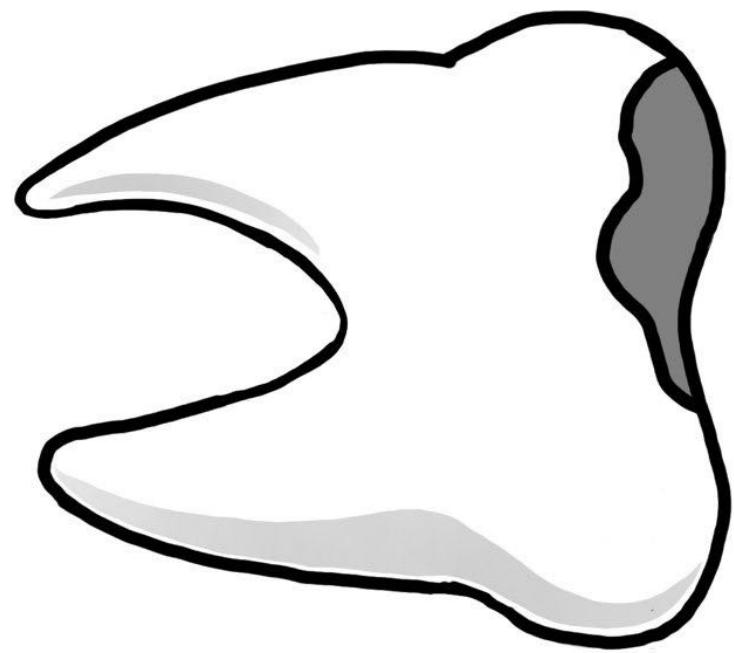
t.me/pepeGprogramming

Made this service
in few weeks

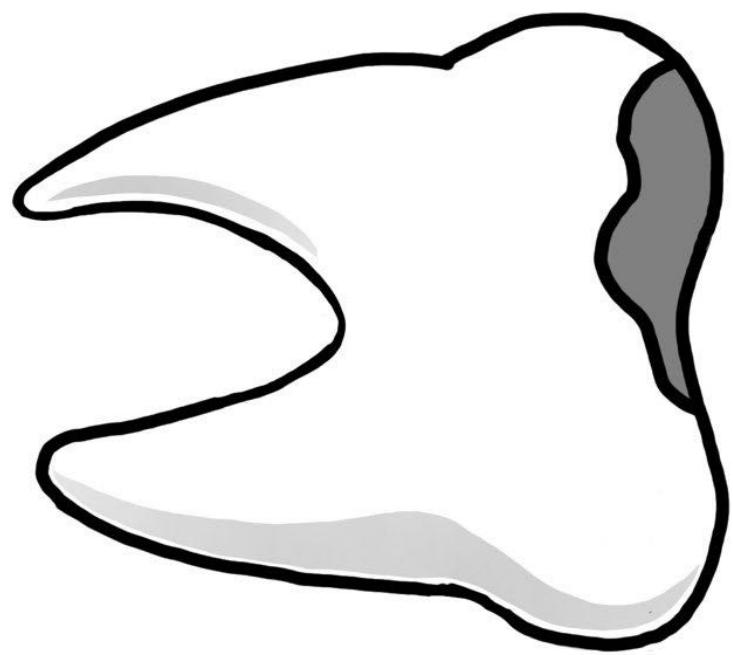
But after that I got a problem

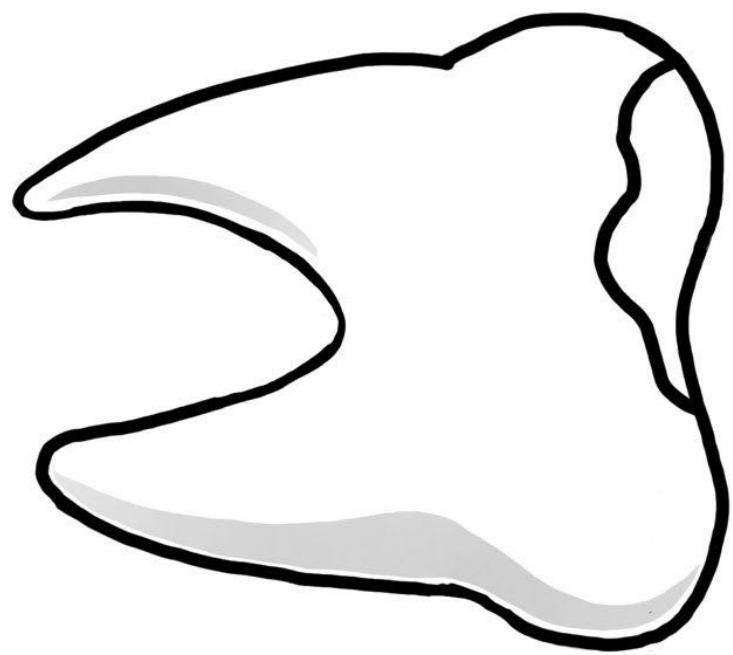


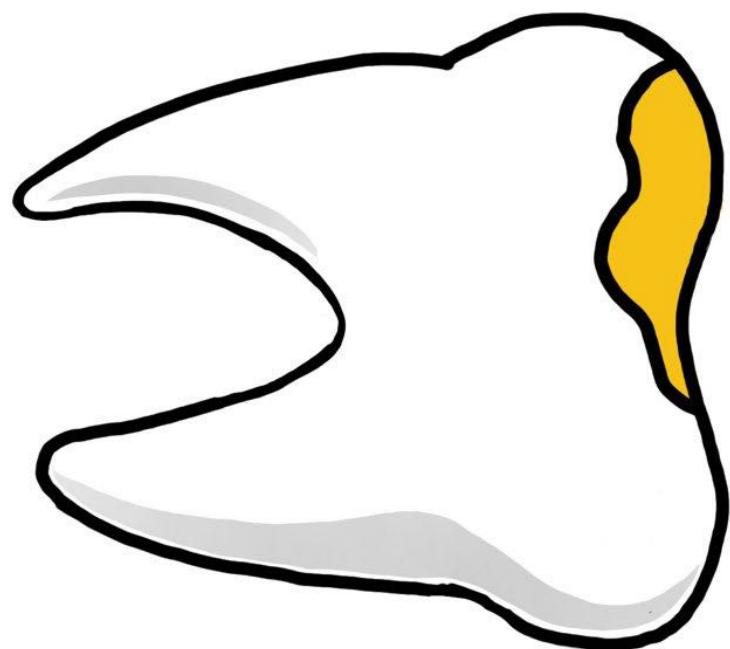


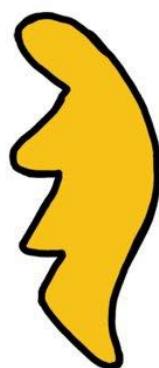
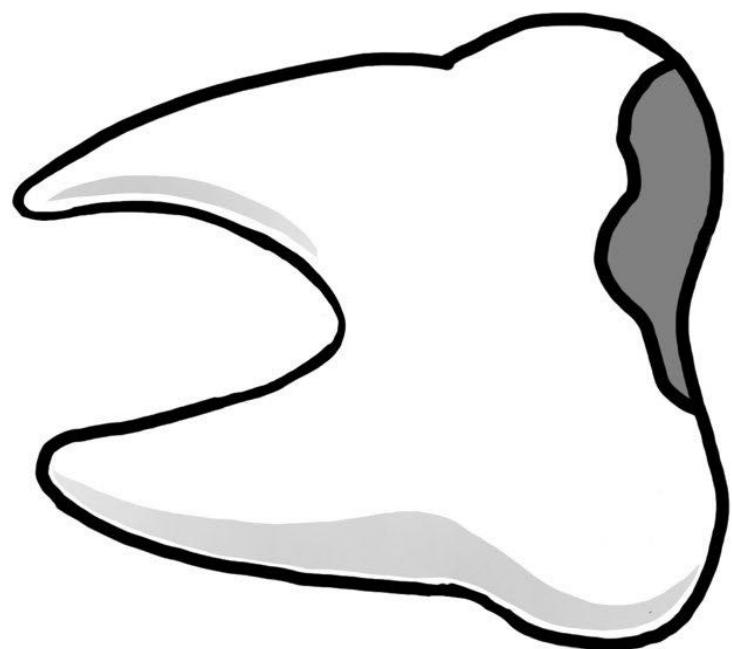






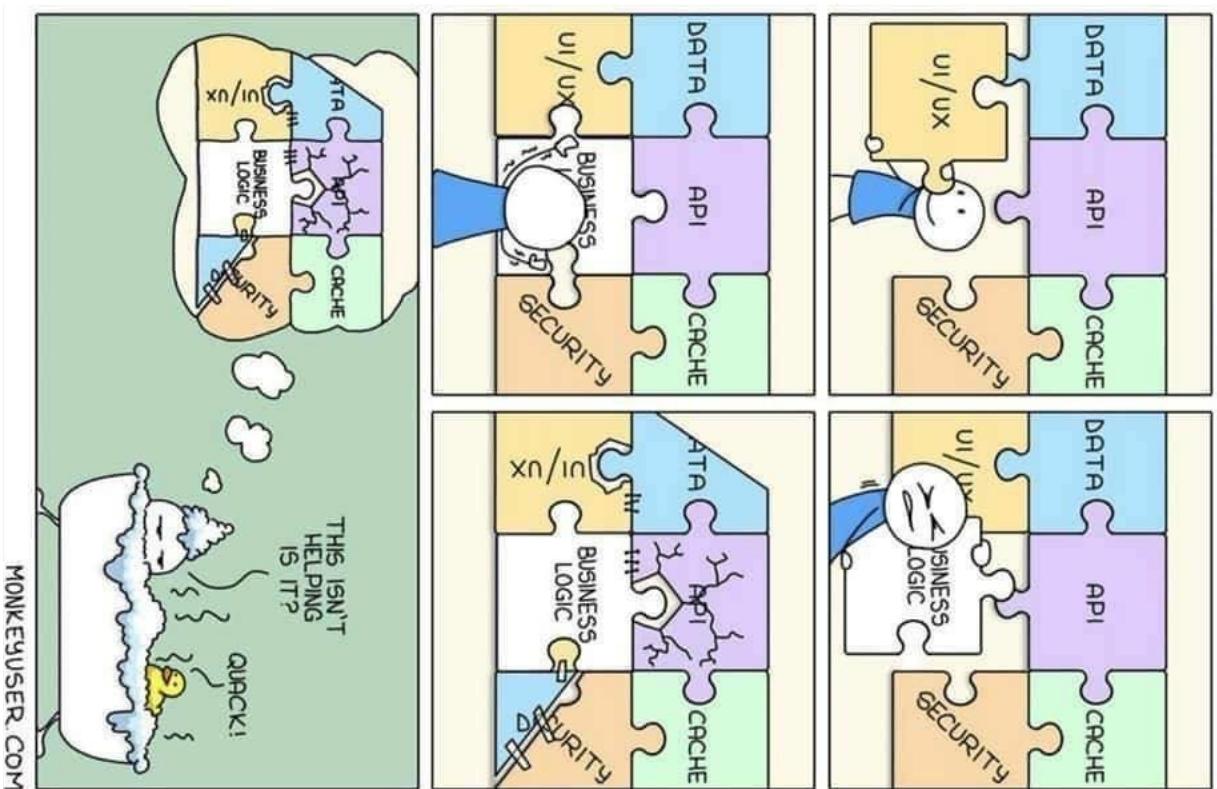






Common problem

ARCHITECTURE



Reverse tactic

isolate prices calls in
the system

```
class Client::V1
  def price(ndc:, pharmacy:, quantity:)
    type = pharmacy_repo.detect_type(pharmacy.to_s)
    price = prices_repo.find_by(ndc, type)

    if price
      Success(total_price: price.cost_for(quantity))
    else
      Failure(:price_not_found)
    end
  end
end
```

```
class Client::V1
  def price(ndc:, pharmacy:, quantity:)
    type = pharmacy_repo.detect_type(pharmacy.to_s)
    price = prices_repo.find_by(ndc, type)

    if price
      Success(total_price: price.cost_for(quantity))
    else
      Failure(:price_not_found)
    end
  end
end
```

```
class Client::V1
  def price(ndc:, pharmacy:, quantity:)
    type = pharmacy_repo.detect_type(pharmacy.to_s)
    price = prices_repo.find_by(ndc, type)

    if price
      Success(total_price: price.cost_for(quantity))
    else
      Failure(:price_not_found)
    end
  end
end
```

Understand consumer
requests and make service
with endpoints for this calls

Make v2 version of client
and switch it in one time

```
class Client::V2
  def price(ndc:, pharmacy:, quantity:)
    @conn
      .get("/prices", { ... })
      .body
      .fmap { |result| { total_price: result[:price] } }
  end
end
```

```
class Client::V2
def price(ndc:, pharmacy:, quantity:)
  @conn
    .get("/prices", { ... })
    .body
    .fmap { |result| { total_price: result[:price] } }
end
end
```

```
class Client::V2
  def price(ndc:, pharmacy:, quantity:)
    @conn
      .get("/prices", { ... })
      .body
      .fmap { |result| { total_price: result[:price] } }
  end
end
```

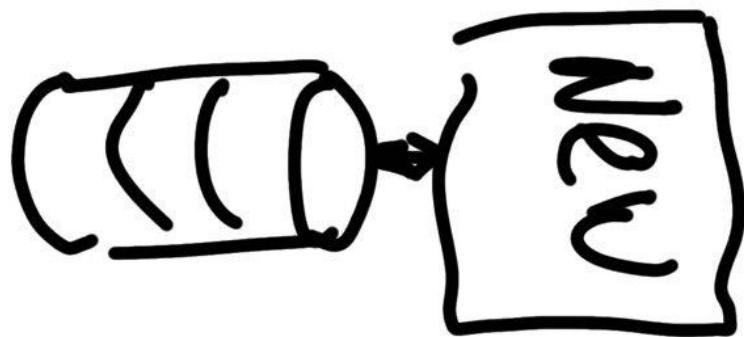
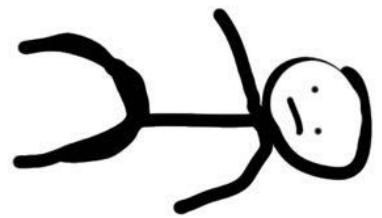
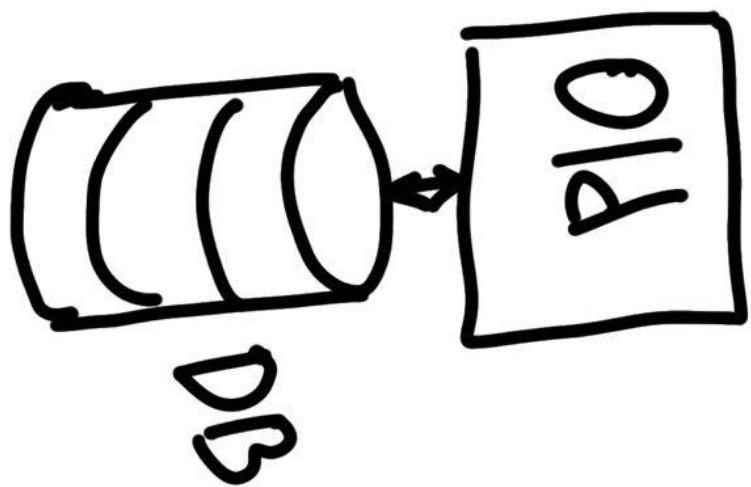
Found unnecessary endpoints

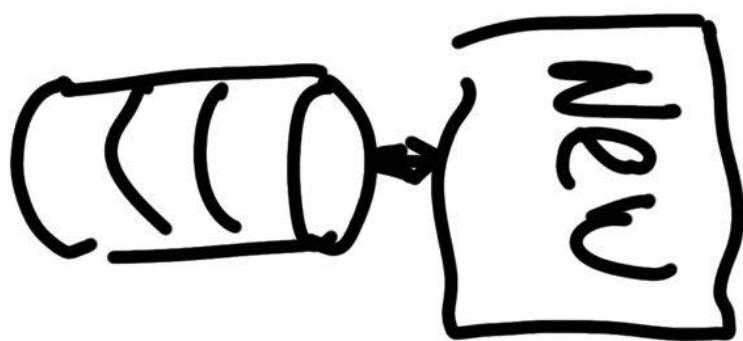
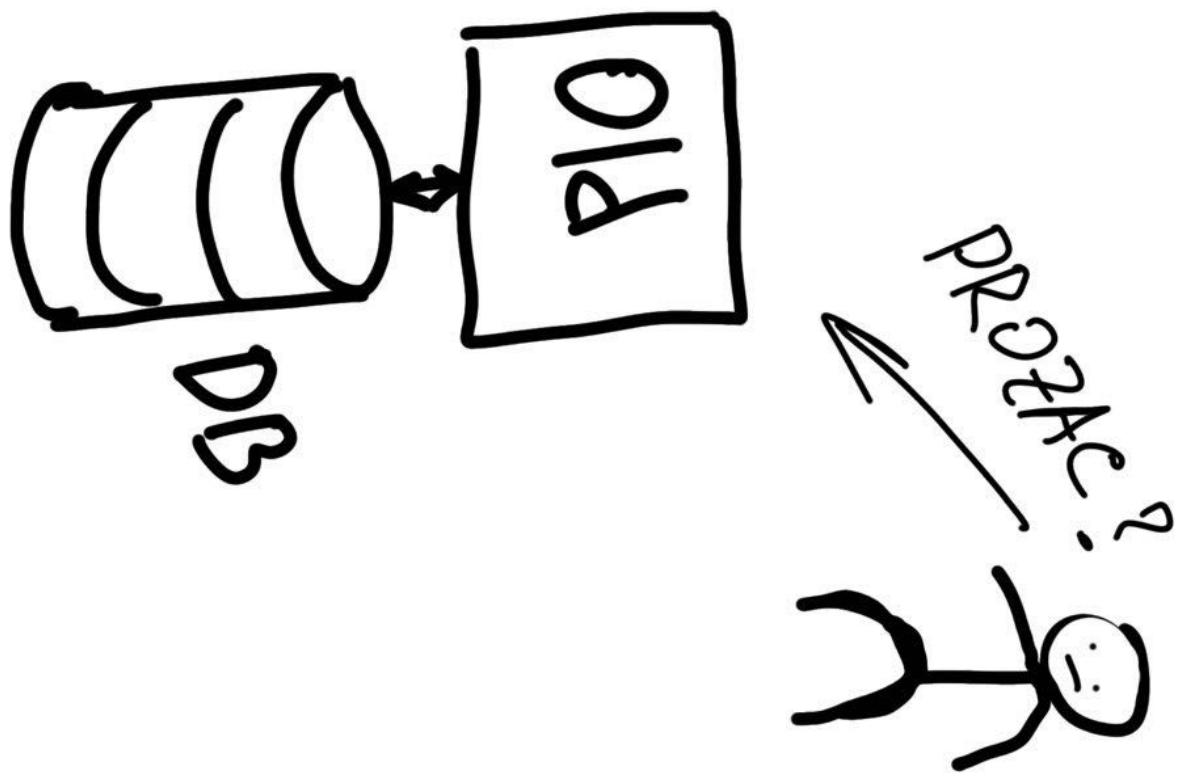
Next problem

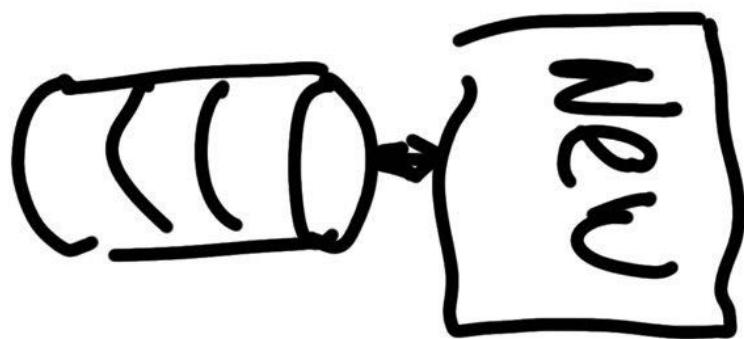
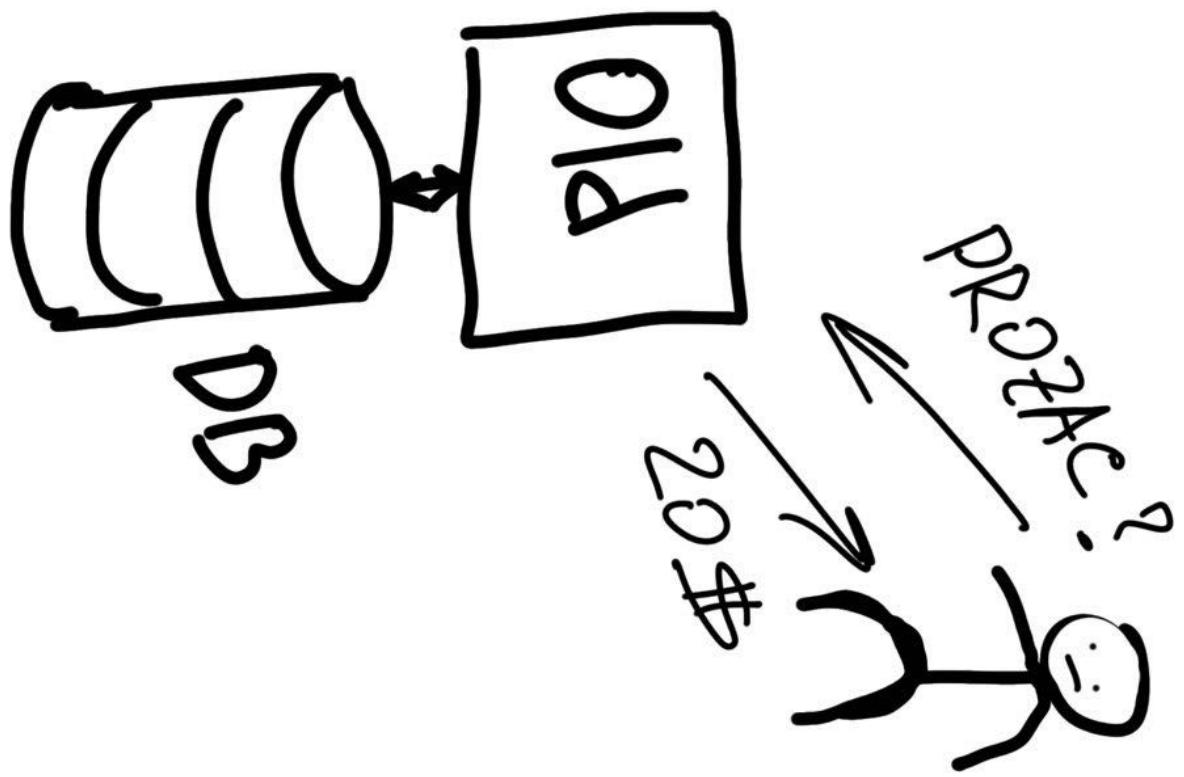
Data synchronisation

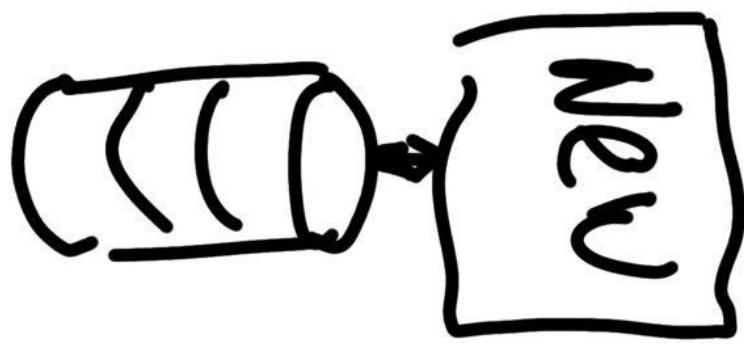
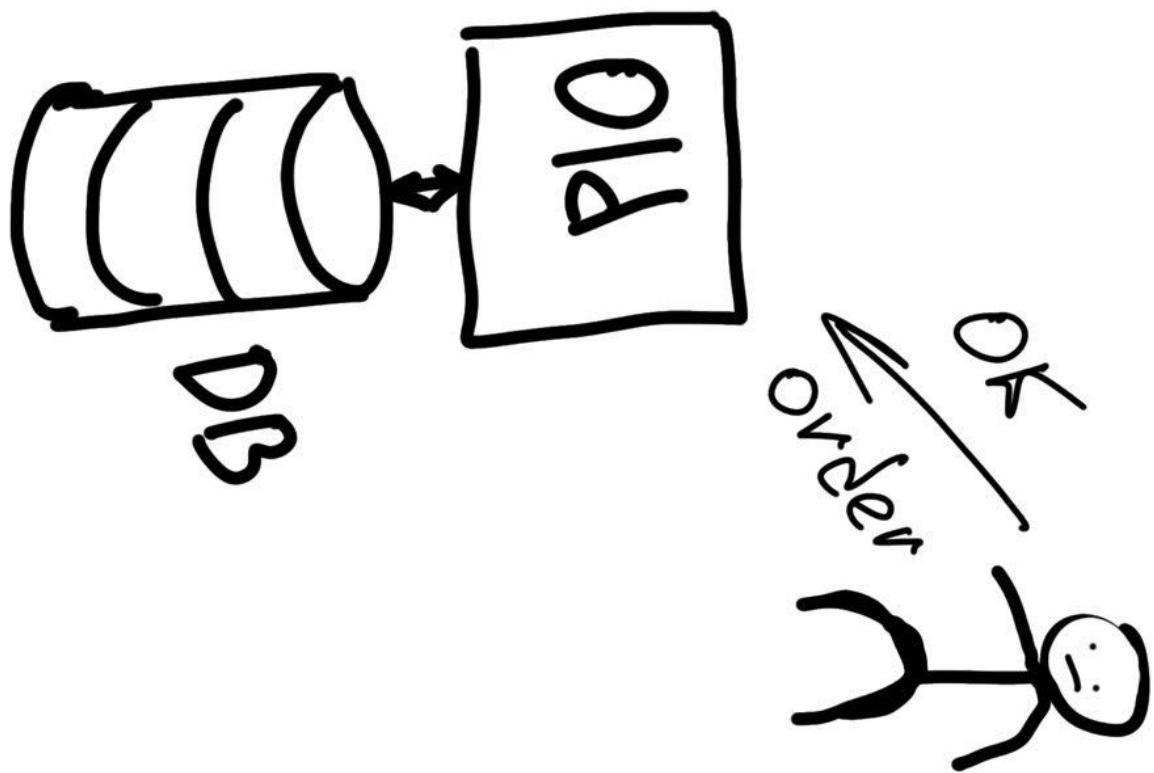
Data sync

Why?

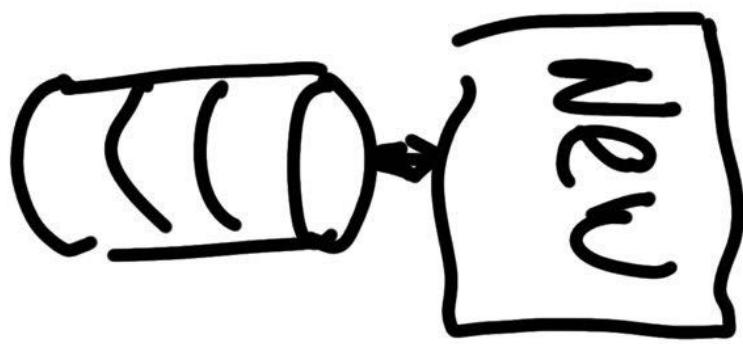
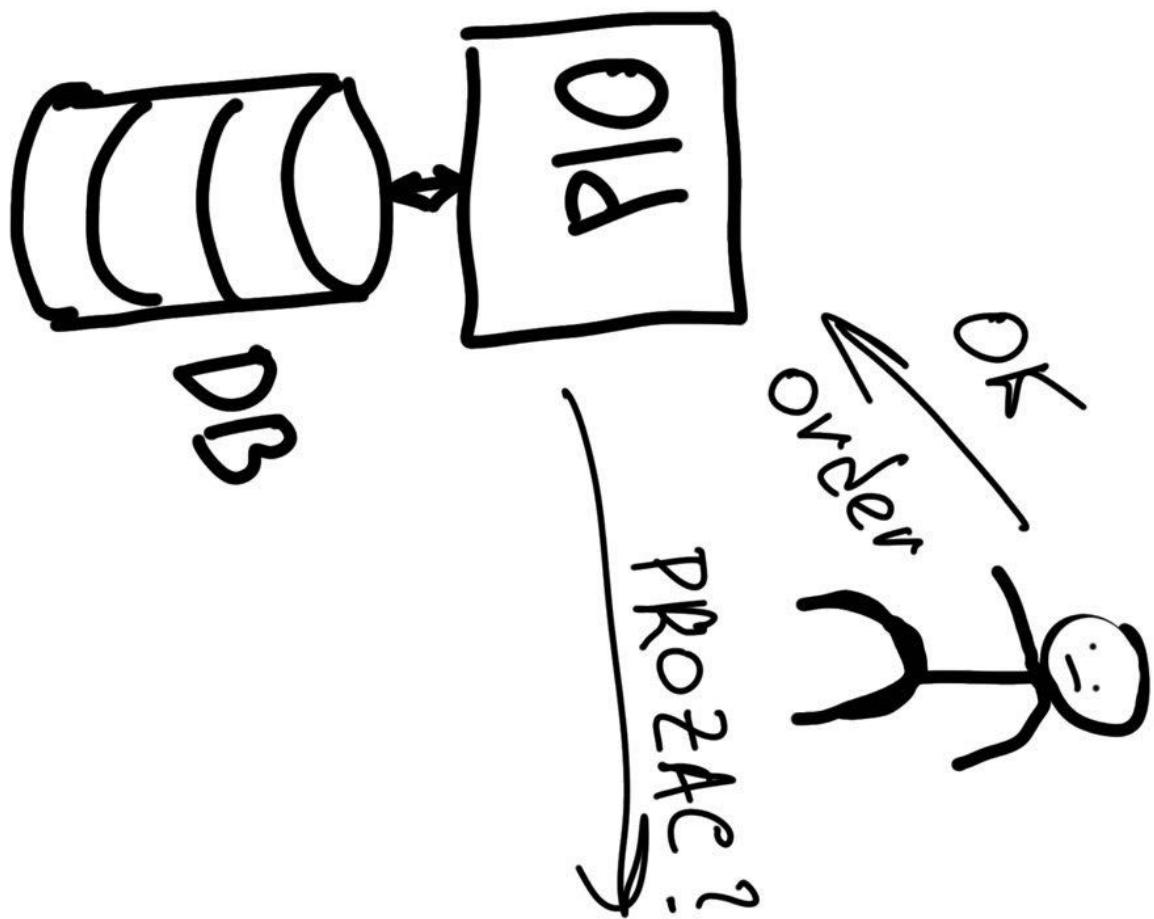


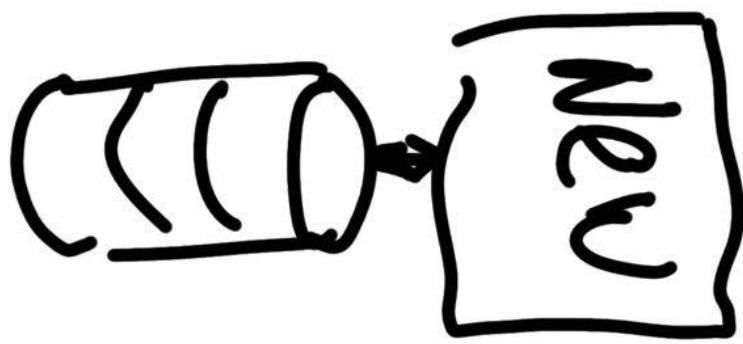
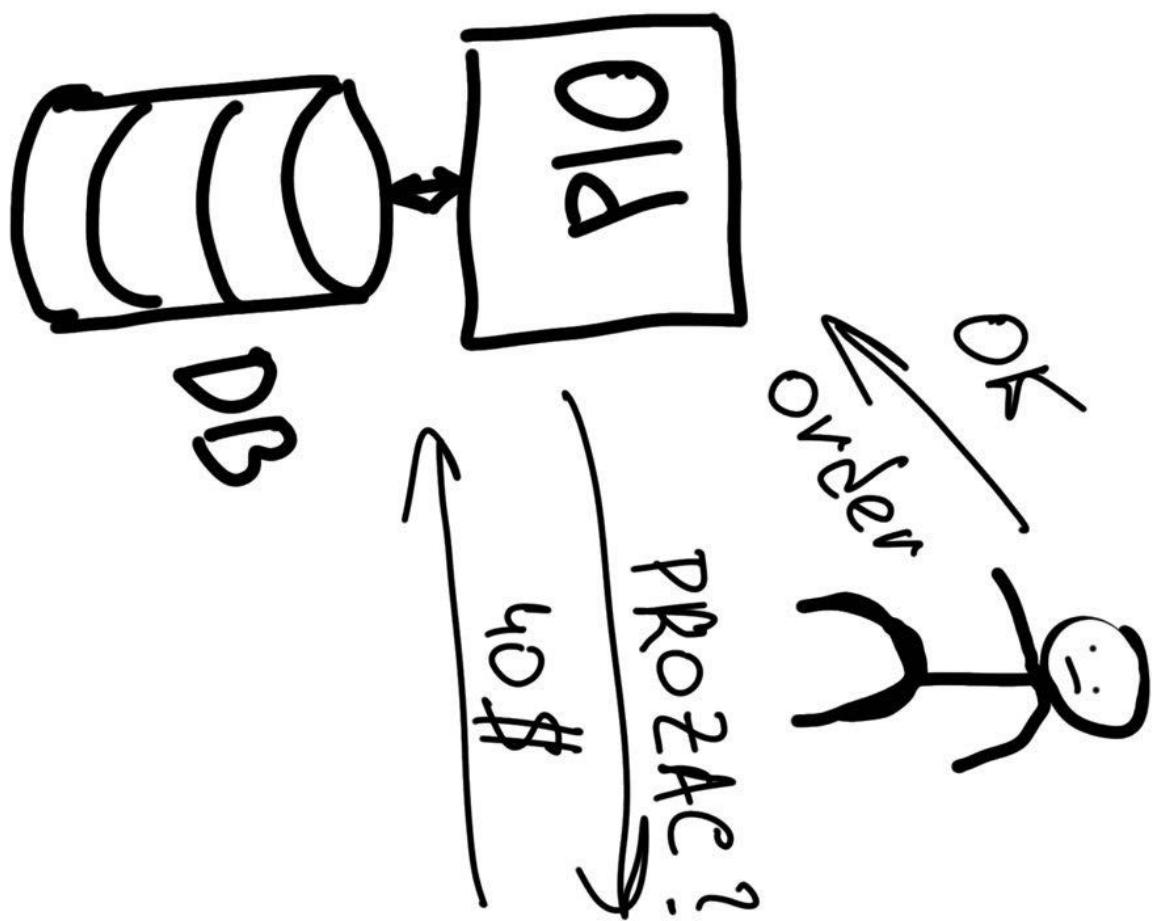


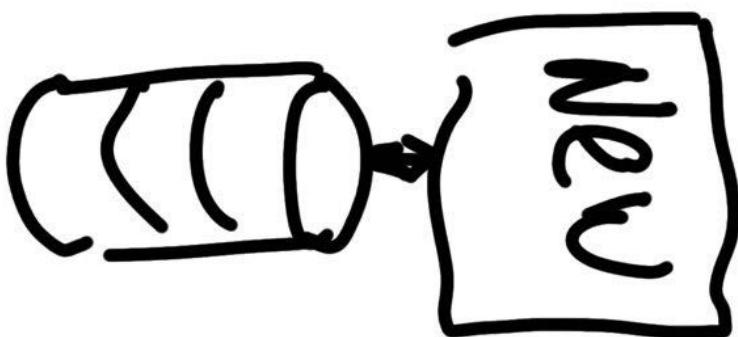
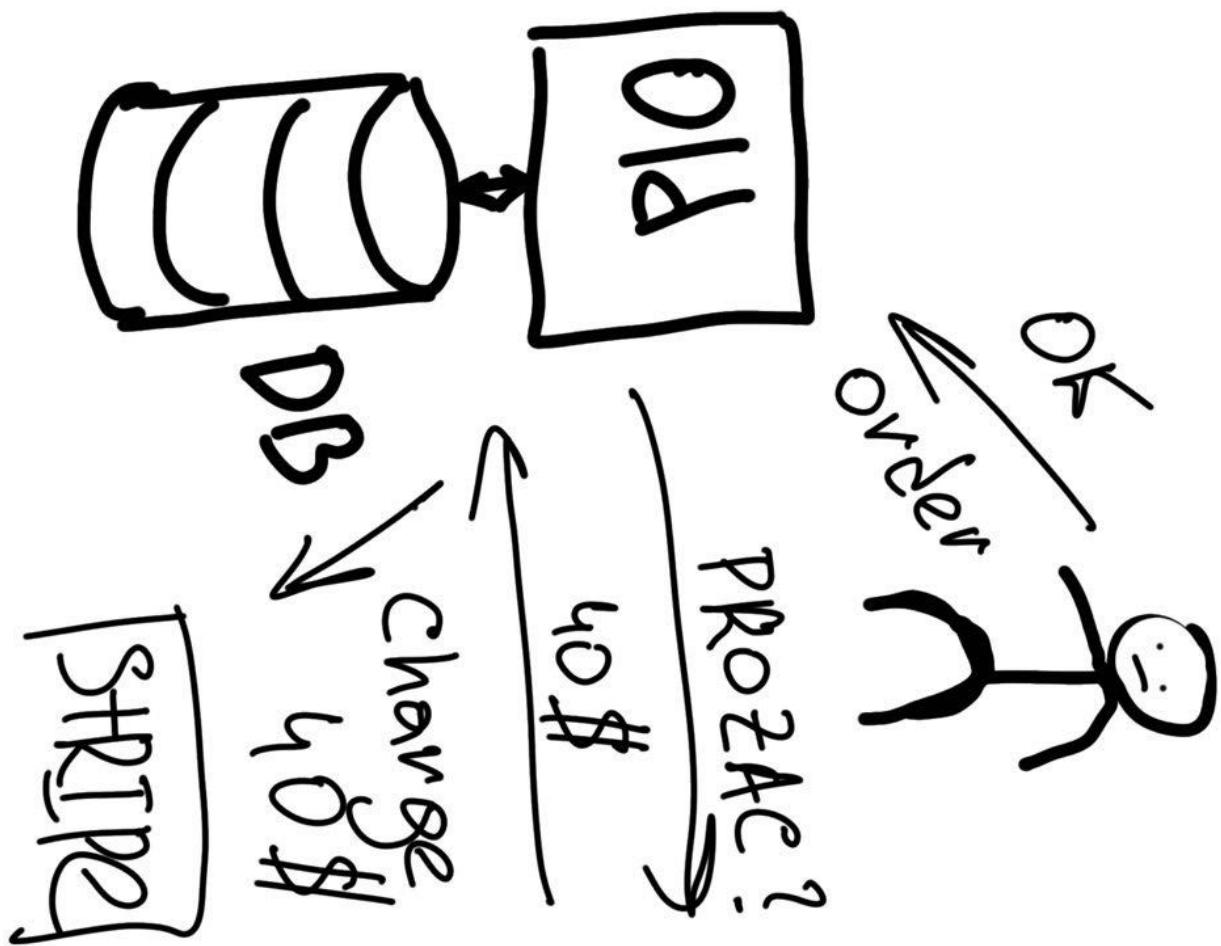


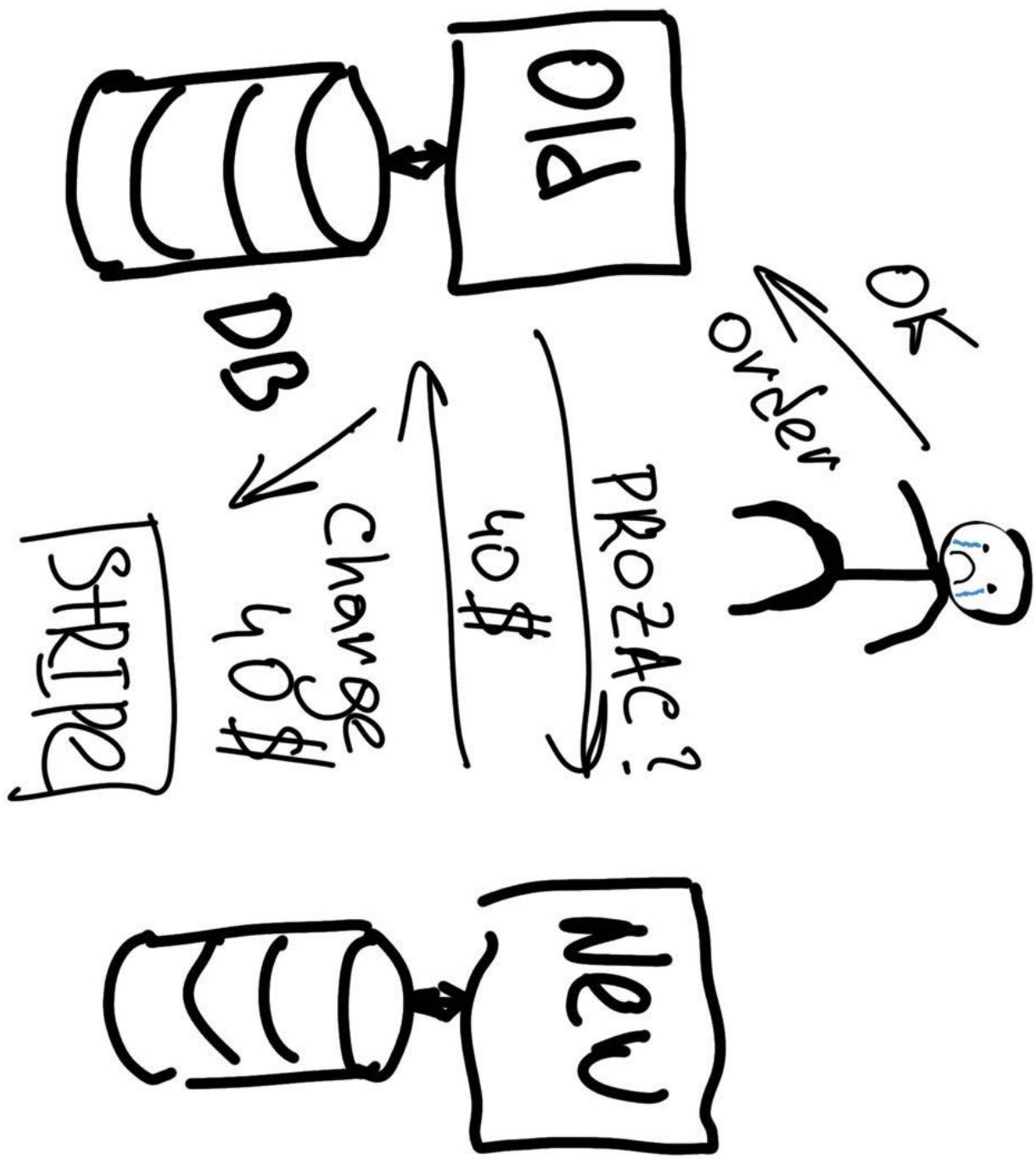


Release time



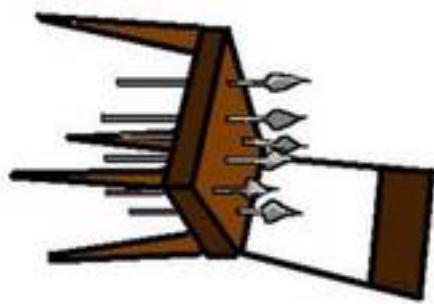
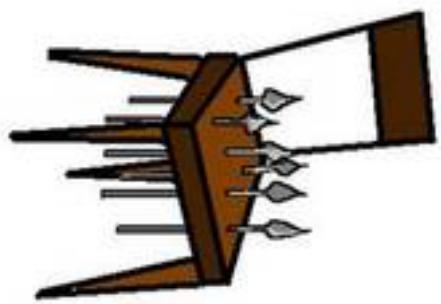






Our solution

Pain



Recalculate data

Make ended prices equal in
different systems

We made a test script for
calling different places and
compare requests

Freeze ETL updates

Pitfalls

Be alone

Thinks a lot

A lot of over engineering
instead flexibility

Got fragility system



No idea how to build
antifragility systems

Principles of Antifragile Software

Martin Monperrus

University of Lille & Inria, France

martin.monperrus@univ-lille1.fr

January 27, 2017

If you know - let's drink



YAGNI

You aren't gonna
need it

BUT

Try to imagine web app
without:

- logging
- data
- error tracking
- etc

You aren't gonna need
it, but you need it



Current time

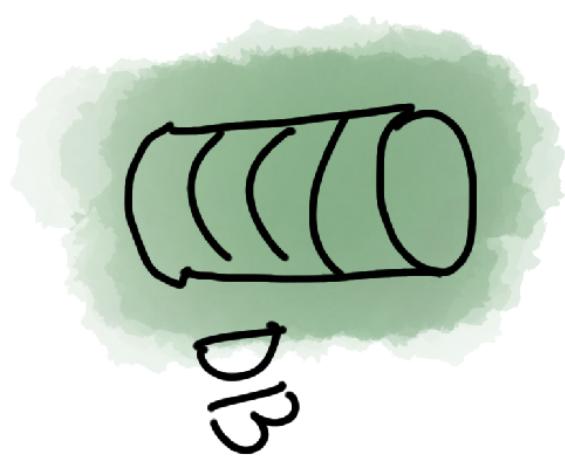
All is okay

That's how I would do it now

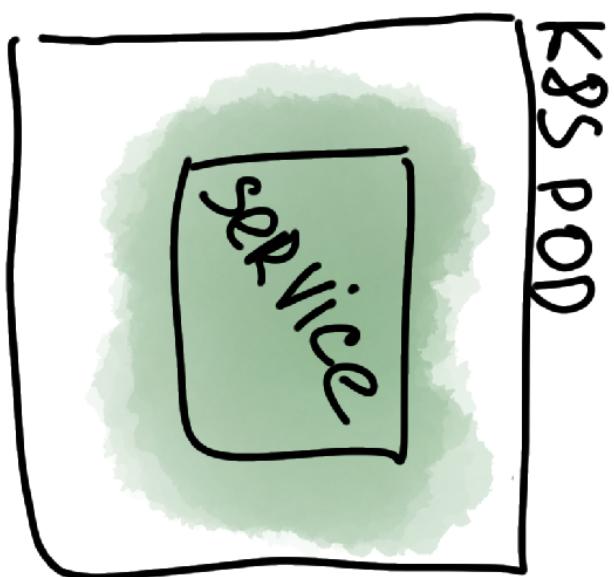
Isolate, after write code

Configure monitoring
and logging

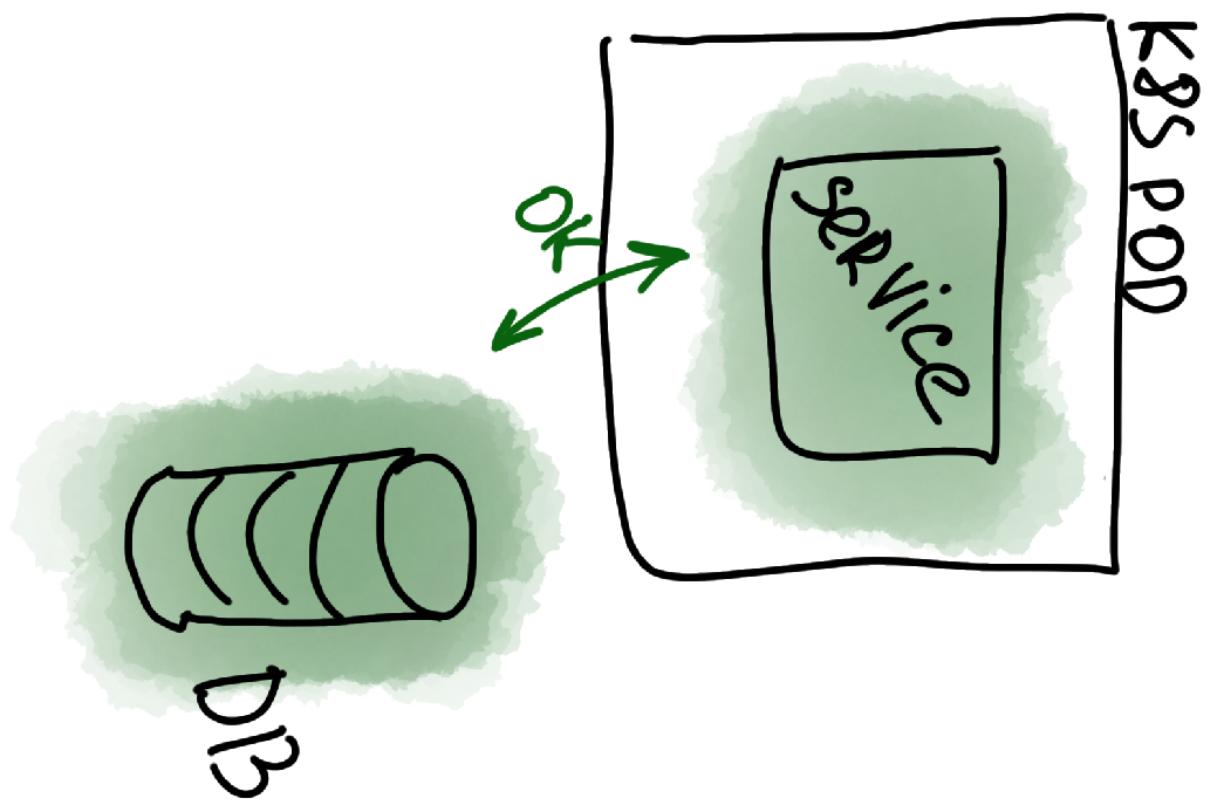
Check DB schema twice

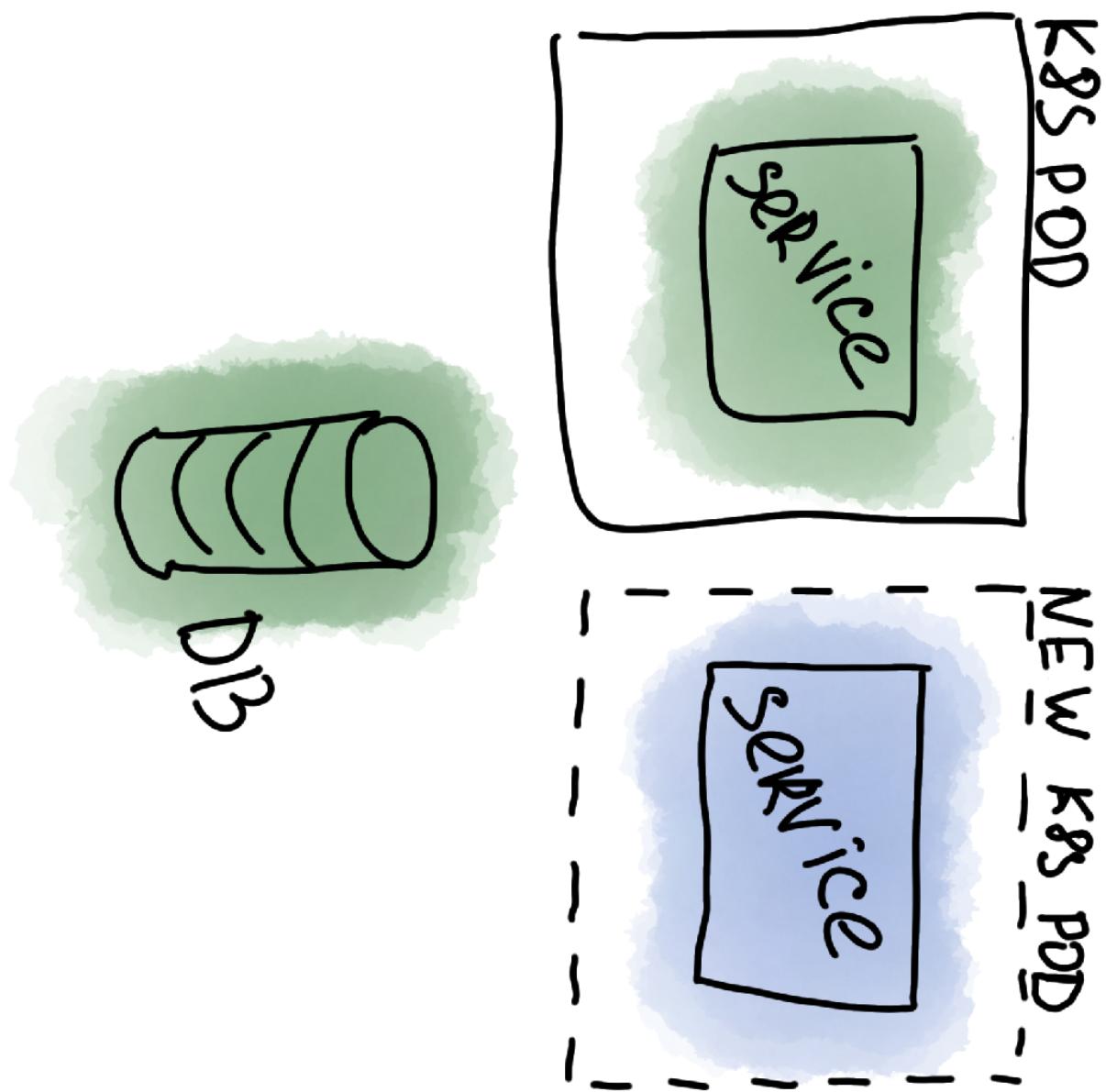


DB

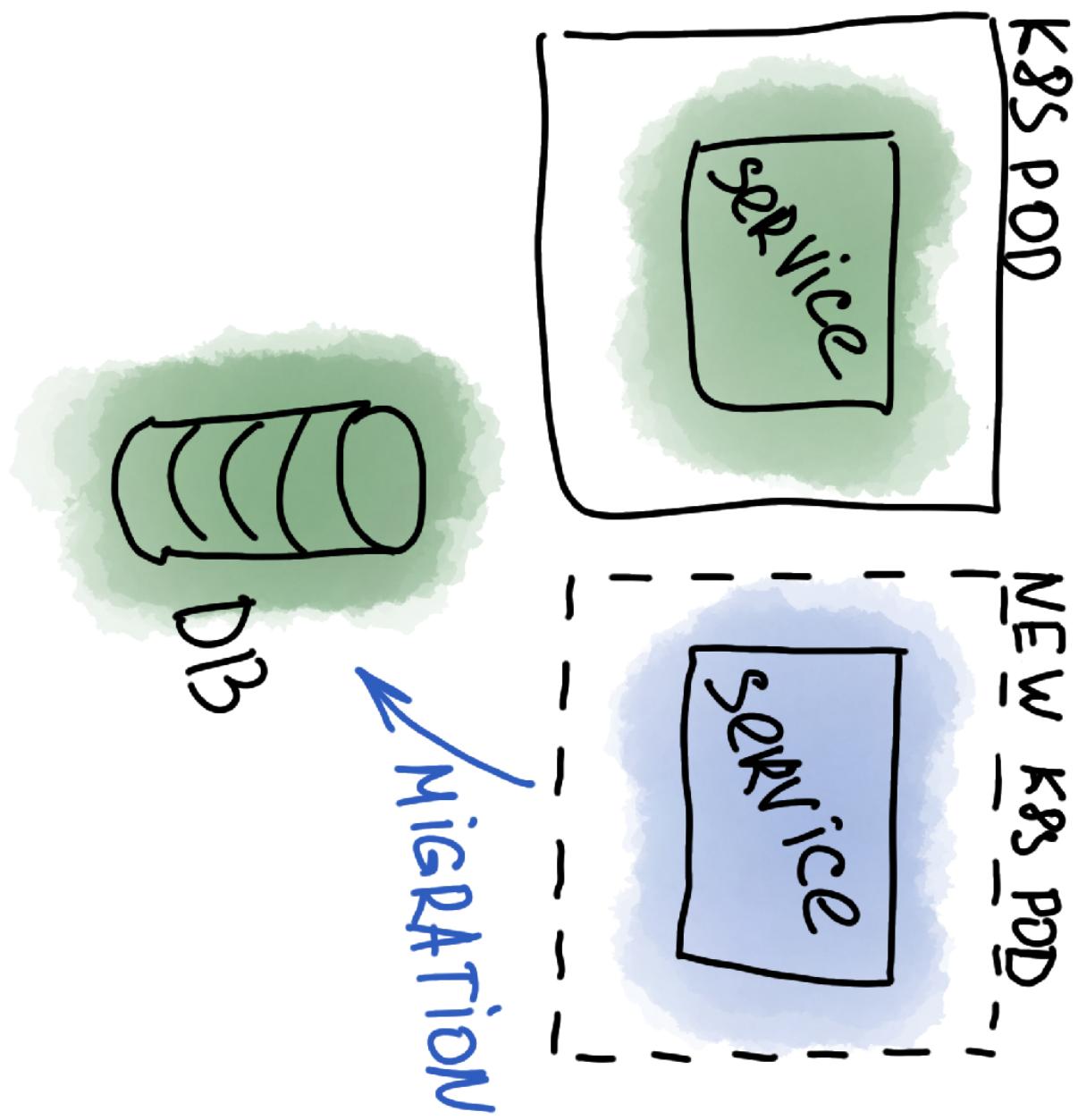


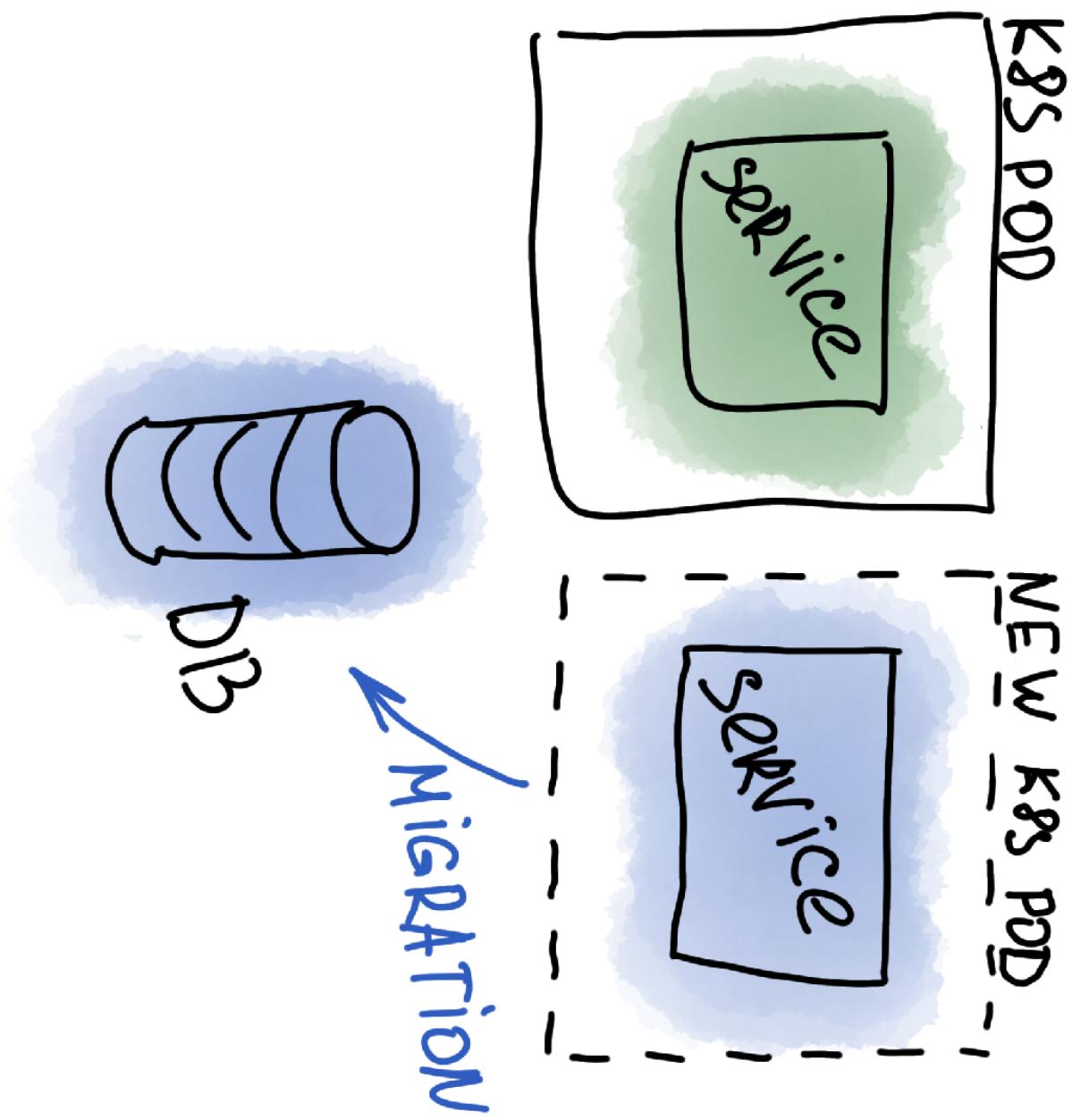
K8S POD

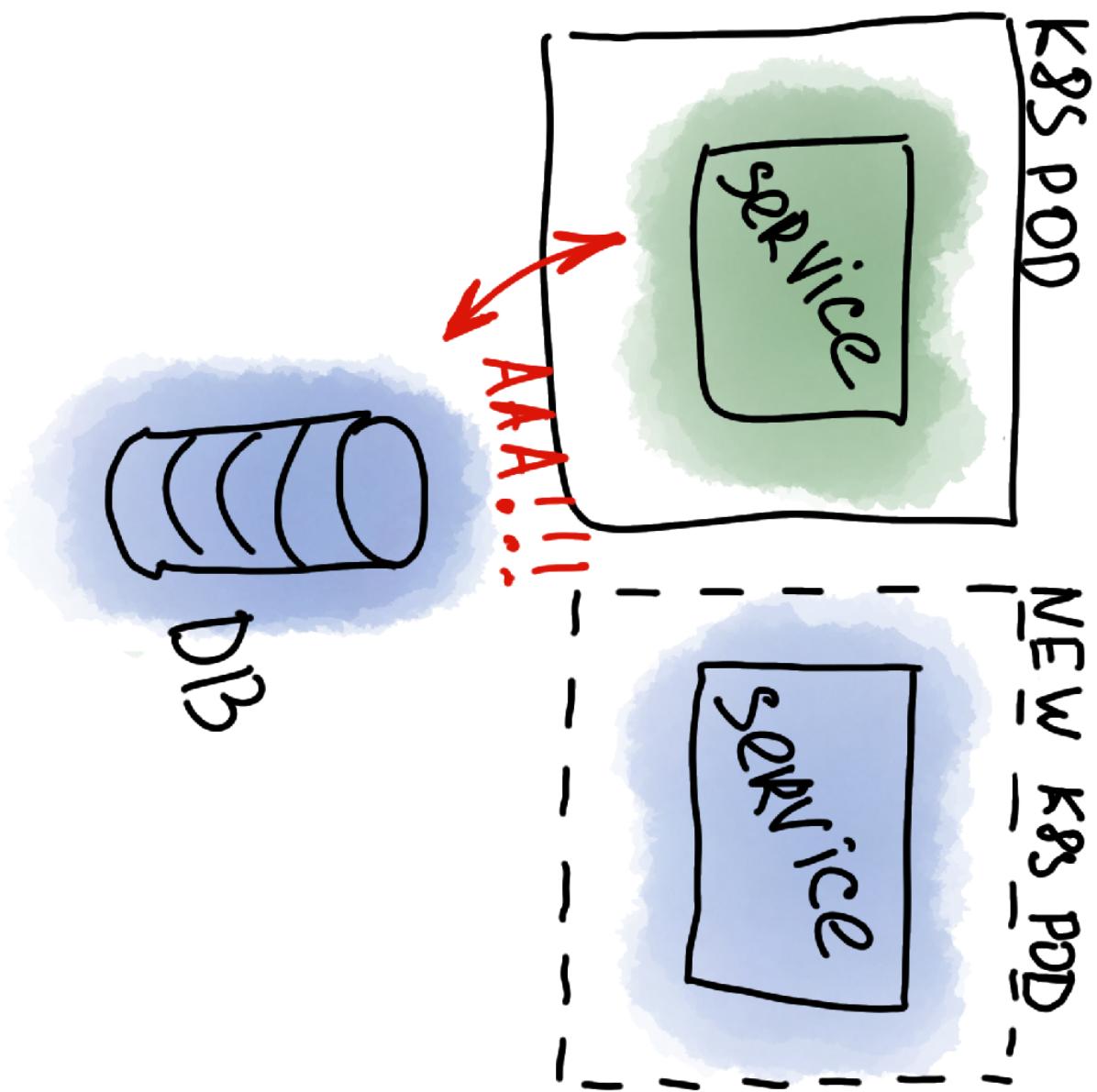




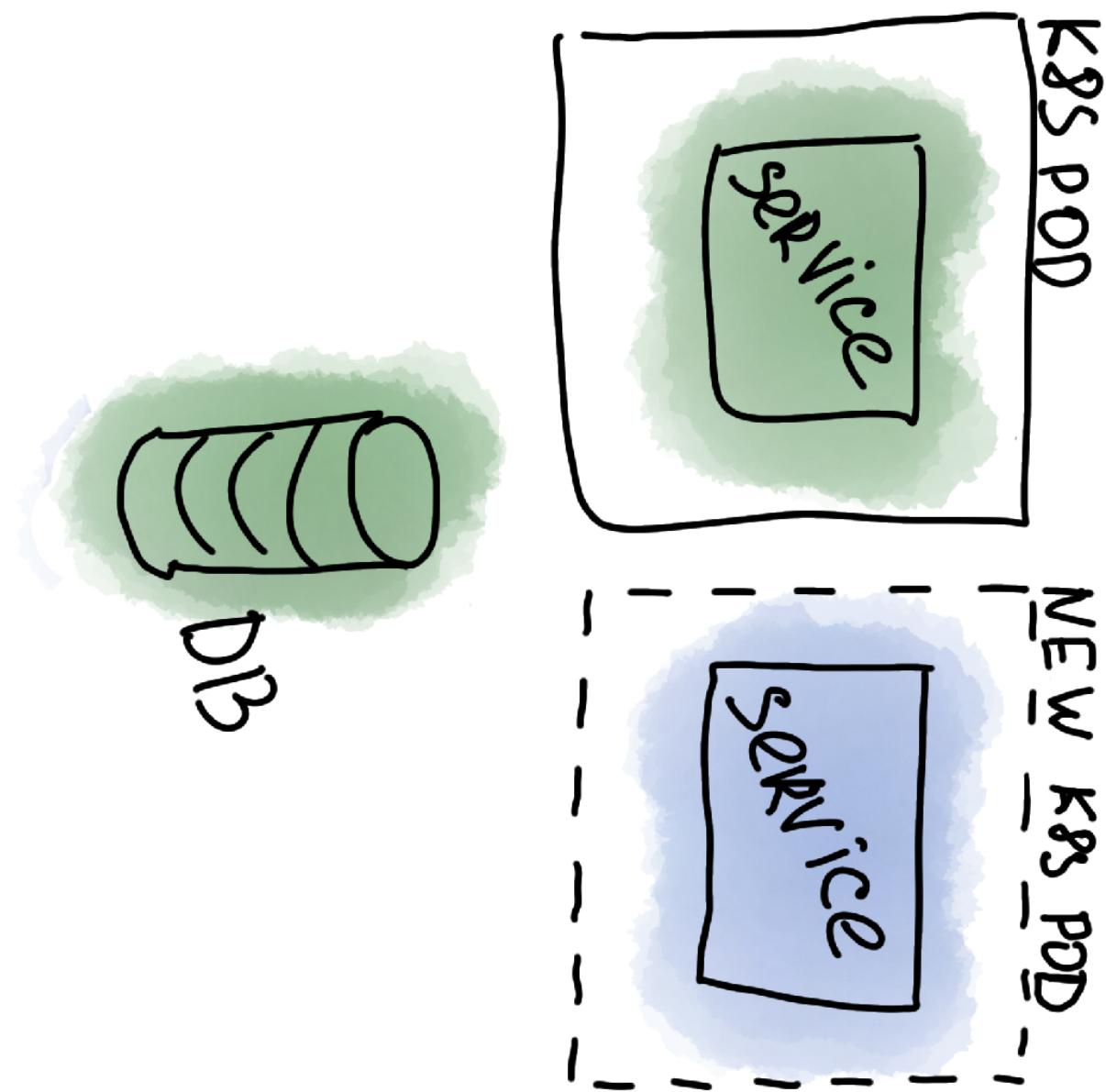
t.me/pepegramming/133

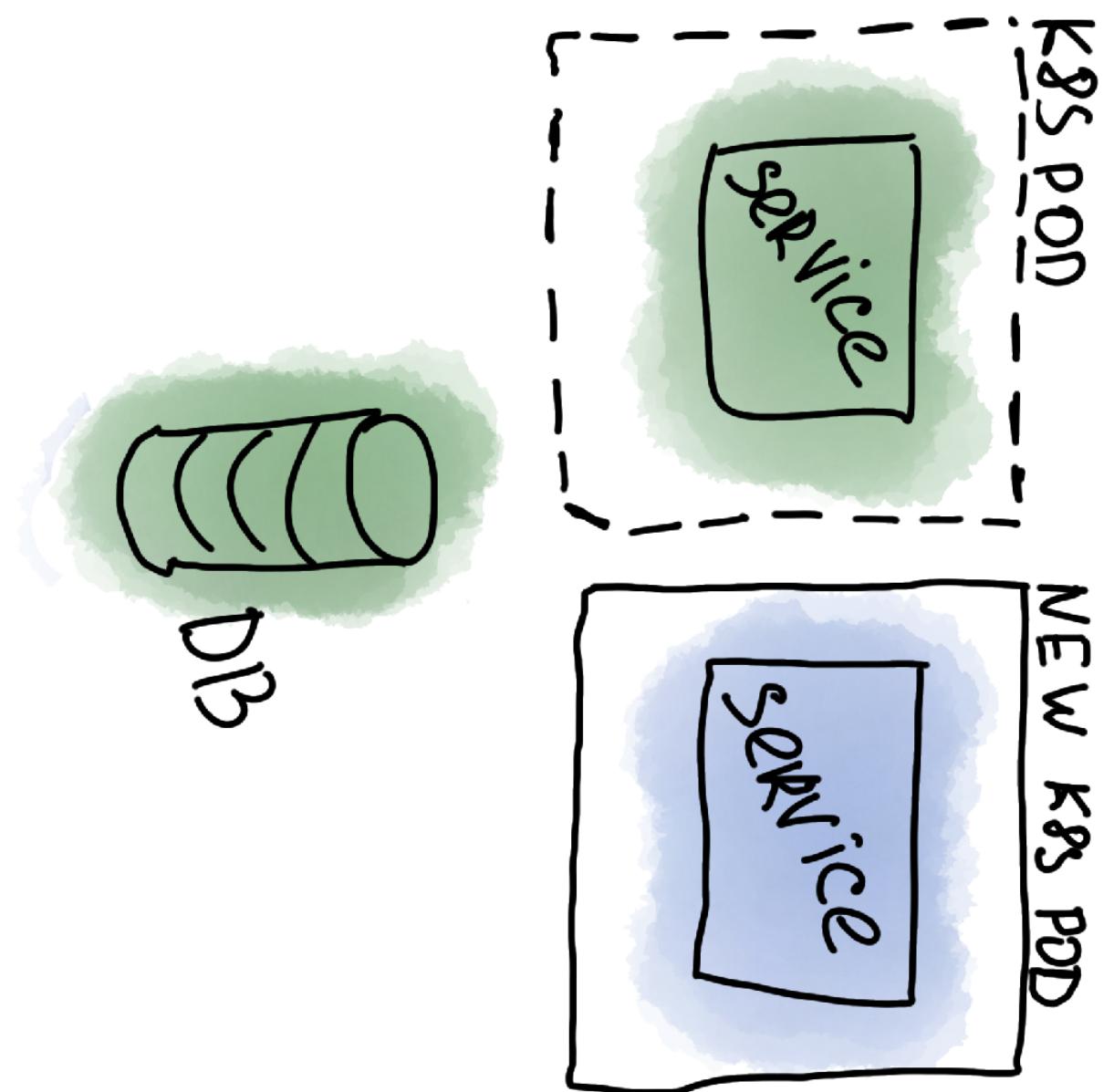


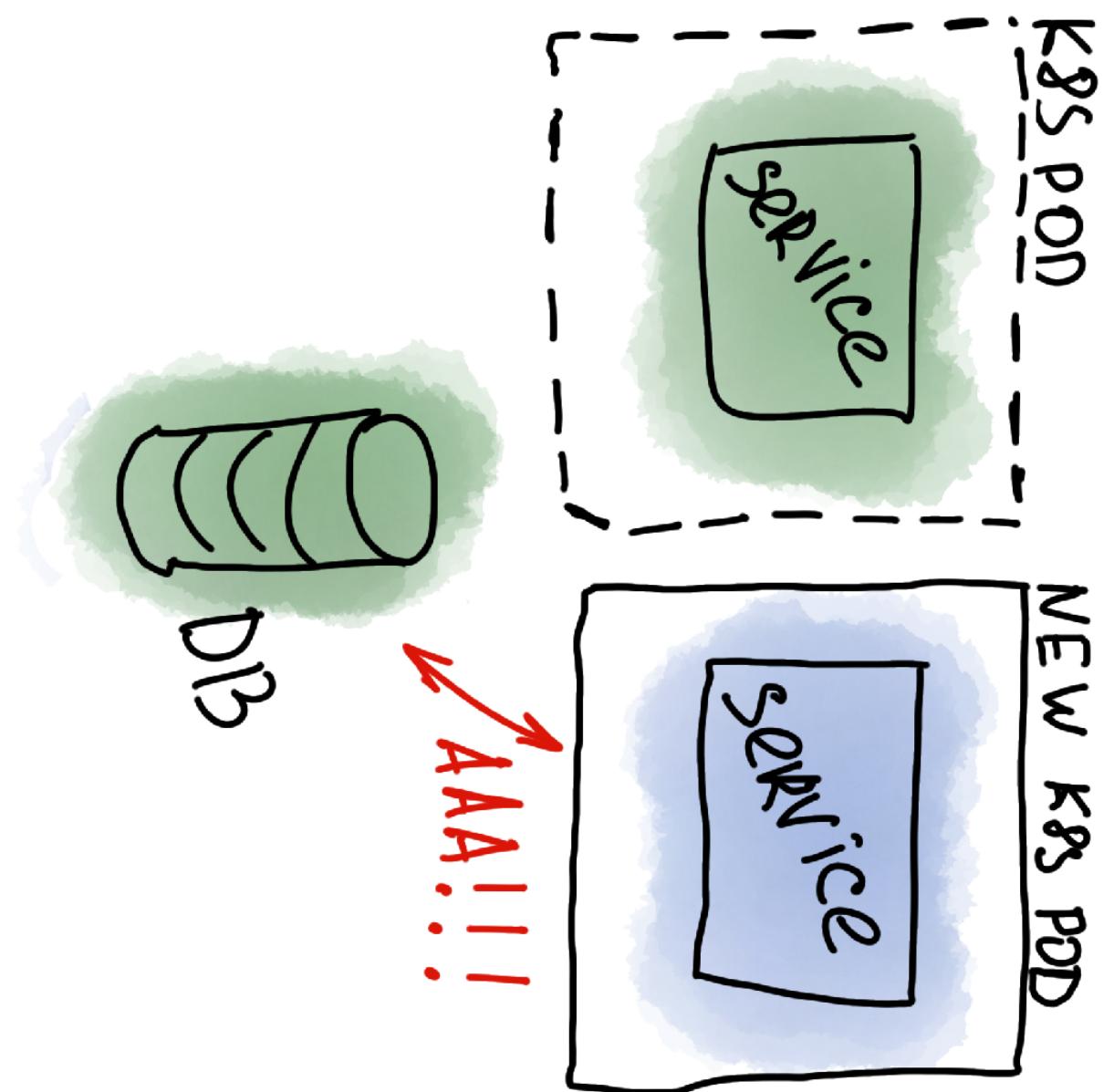


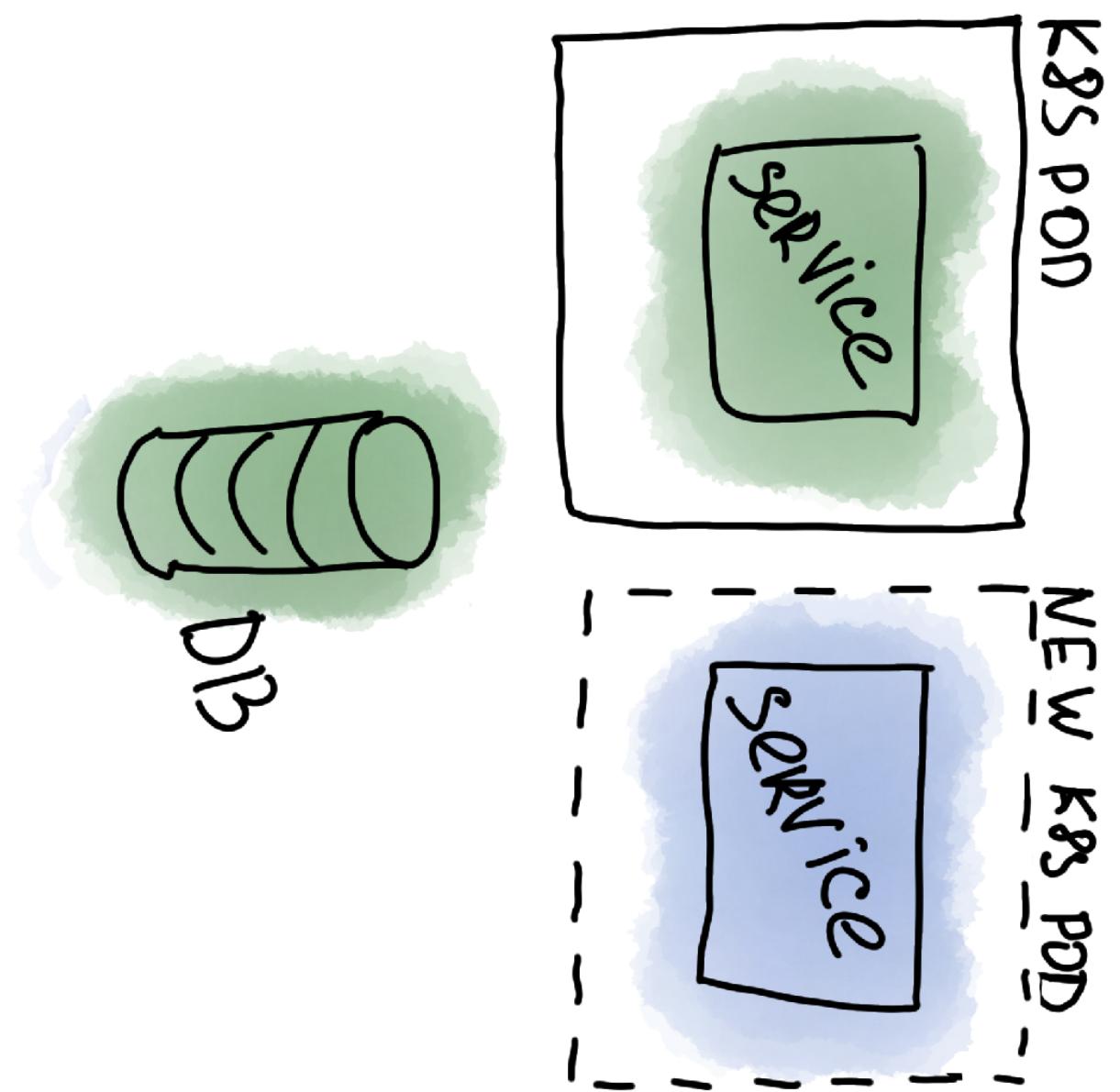


1.06.20











Conclusions

Data layer is really important

Requirements can break
ideal architecture

Requirements can
happen but it's okay

Best what you can - isolate everything

Silver bullet is myth

Bad solution is solution too

Resources

O'REILLY®

Designing Data-Intensive Applications

THE BIG IDEAS BEHIND RELIABLE, SCALABLE,
AND MAINTAINABLE SYSTEMS



Martin Kleppmann

@martinkl

dataintensive.net

Netflix architecture talk



2|rogramming

777 subscribers

info

Грустно о программировании. Все проблемы сюда: [@davydovanton](https://t.me/davydovanton)

Thank you ❤

github.com/davydovanton

twitter.com/anton_davydov

davydovanton.com