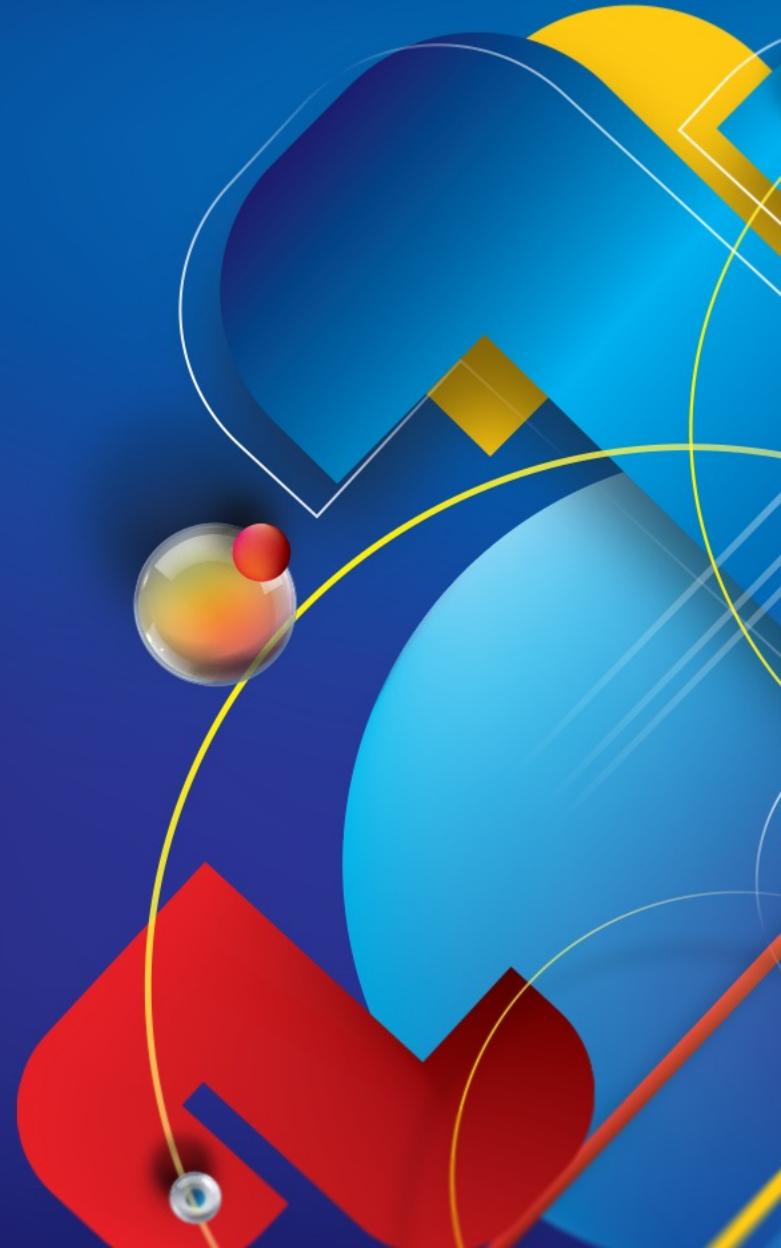


Двусторонний websocket-роутинг

Владислав Лаухин
Аникин Денис



Moscow ★
Python Conf++
2021



Владислав Лаухин

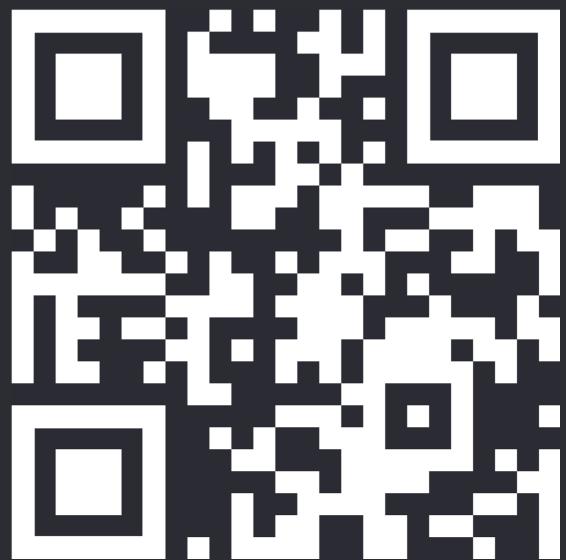
- разработчик в команде Chat, Raiffeisenbank
- пилю чат и чатбота
- развиваю с коллегами коммьюнити питонистов Райфа
- питон, разработка, девопс



Аникин Денис

Вдруг кому-то важно кто я такой

- team lead в команде Chat, Raiffeisenbank
- community lead в Python Community
- fullstack: разрабатываю на python и typescript
- занимаюсь развитием DevOps практик
- мой сайт: <https://xfenix.ru/>



**В начале опишем
контекст**

Что мы делали?

— омниканальный чат со своей админкой

Что мы делали?

- омниканальный чат со своей админкой
- систему, с которой работают и операторы и клиенты

Что мы делали?

- омниканальный чат со своей админкой
- систему, с которой работают и операторы и клиенты
- систему, в которой много вебсокет соединений, которые эксплуатируются в 2 направлениях

Что мы делали?

- омниканальный чат со своей админкой
- систему, с которой работают и операторы и клиенты
- систему, в которой много вебсокет соединений, которые эксплуатируются в 2 направлениях
- нам нужно было держать нагрузку и достичь отказоустойчивости

Цели презентации

— В начале нашей разработки, в интернете не было информации по проблеме. Поэтому мы захотели поделиться

Цели презентации

- В начале нашей разработки, в интернете не было информации по проблеме. Поэтому мы захотели поделиться
- Информации и сейчас немного

Цели презентации

- В начале нашей разработки, в интернете не было информации по проблеме. Поэтому мы захотели поделиться
- Информации и сейчас немного
- Хотелось показать, что в среде python разработки существуют не только REST'ы и CRUD'ы

Перейдём к делу

Шаг 0: готовые решения

Пытаемся понять — подходят ли они нам?



Mercure:

- нет четкой уверенности, что
http2 + sse > websocket
- менее популярное решение
- меньше на слуху
- у нас энтерпрайз ☺ (лицензии?)

Шаг 0: готовые решения

Пытаемся понять — подходят ли они нам?



Mercure:

- нет четкой уверенности, что
 http2 + sse > websocket
- менее популярное решение
- меньше на слуху
- у нас энтерпрайз ☺ (лицензии?)



Centrifugo:

- соединение в одном направлении
- канал «вниз» нужно делать самим

Ну раз нет, то...

iStock
by Getty Images™

iStock
by Getty Images™

iStock
by Getty Images™

iStock
by Getty Images™



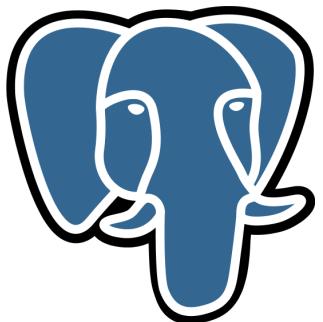
iStock
by Getty Images™

iStock
by Getty Images™

iStock
by Getty Images™

iStock
by Getty Images™

Наш стек



Больше о проблеме вебсокетов

применительно к нашим условиям

2 направления

Роутинг необходимо
осуществлять в двух
направлениях

Больше о проблеме вебсокетов

применительно к нашим условиям

2 направления

Роутинг необходимо
осуществлять в двух
направлениях



Роутинг «вверх» —
сложная проблема
(с ней мы и боремся в
этом докладе)

Больше о проблеме вебсокетов

применительно к нашим условиям

2
направления

Роутинг необходимо
осуществлять в двух
направлениях



Роутинг «вверх» —
сложная проблема
(с ней мы и боремся в
этом докладе)



Решение необходимо
горизонтально
масштабировать

**Но для начала поговорим
как мы уже решали эту
проблему в другом
продукте**

Подход к проблеме 1.0

Что за продукт связан с «проблемой 1.0»?

— чат-бот. Его задача — помогать клиентам решать типовые проблемы

Что за продукт связан с «проблемой 1.0»?

- чат-бот. Его задача — помогать клиентам решать типовые проблемы
- обслуживает весь текстовый канал

Что за продукт связан с «проблемой 1.0»?

- чат-бот. Его задача — помогать клиентам решать типовые проблемы
- обслуживает весь текстовый канал
- частично обслуживает голос

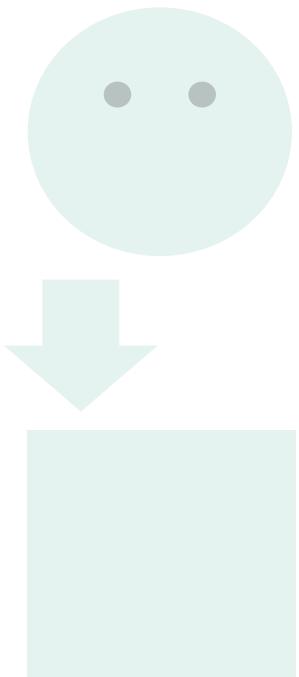
Что за продукт связан с «проблемой 1.0»?

- чат-бот. Его задача — помогать клиентам решать типовые проблемы
- обслуживает весь текстовый канал
- частично обслуживает голос
- в будущем будет сам звонить клиентам

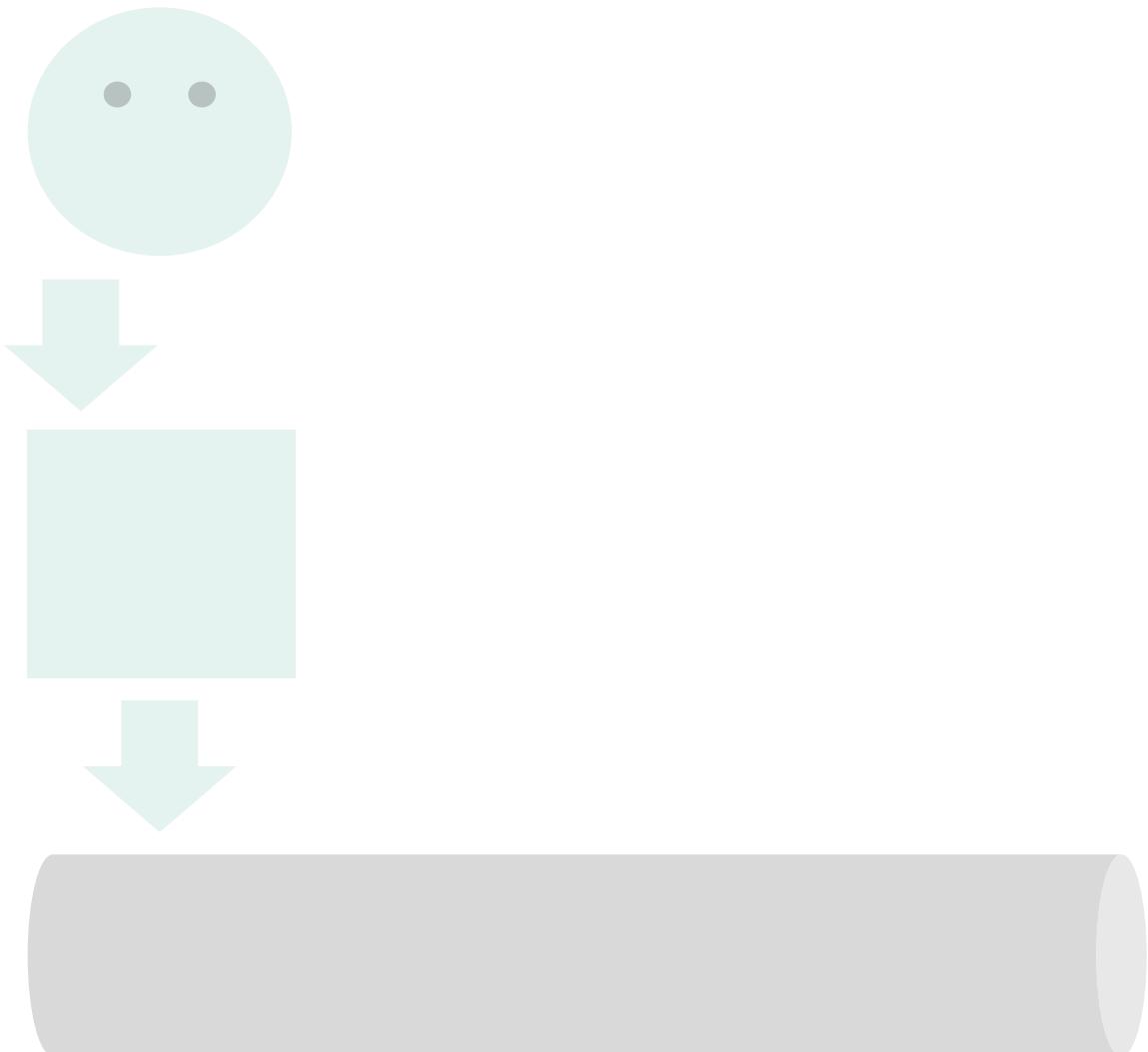
Что за продукт связан с «проблемой 1.0»?

- чат-бот. Его задача — помогать клиентам решать типовые проблемы
- обслуживает весь текстовый канал
- частично обслуживает голос
- в будущем будет сам звонить клиентам
- имеет много интеграций

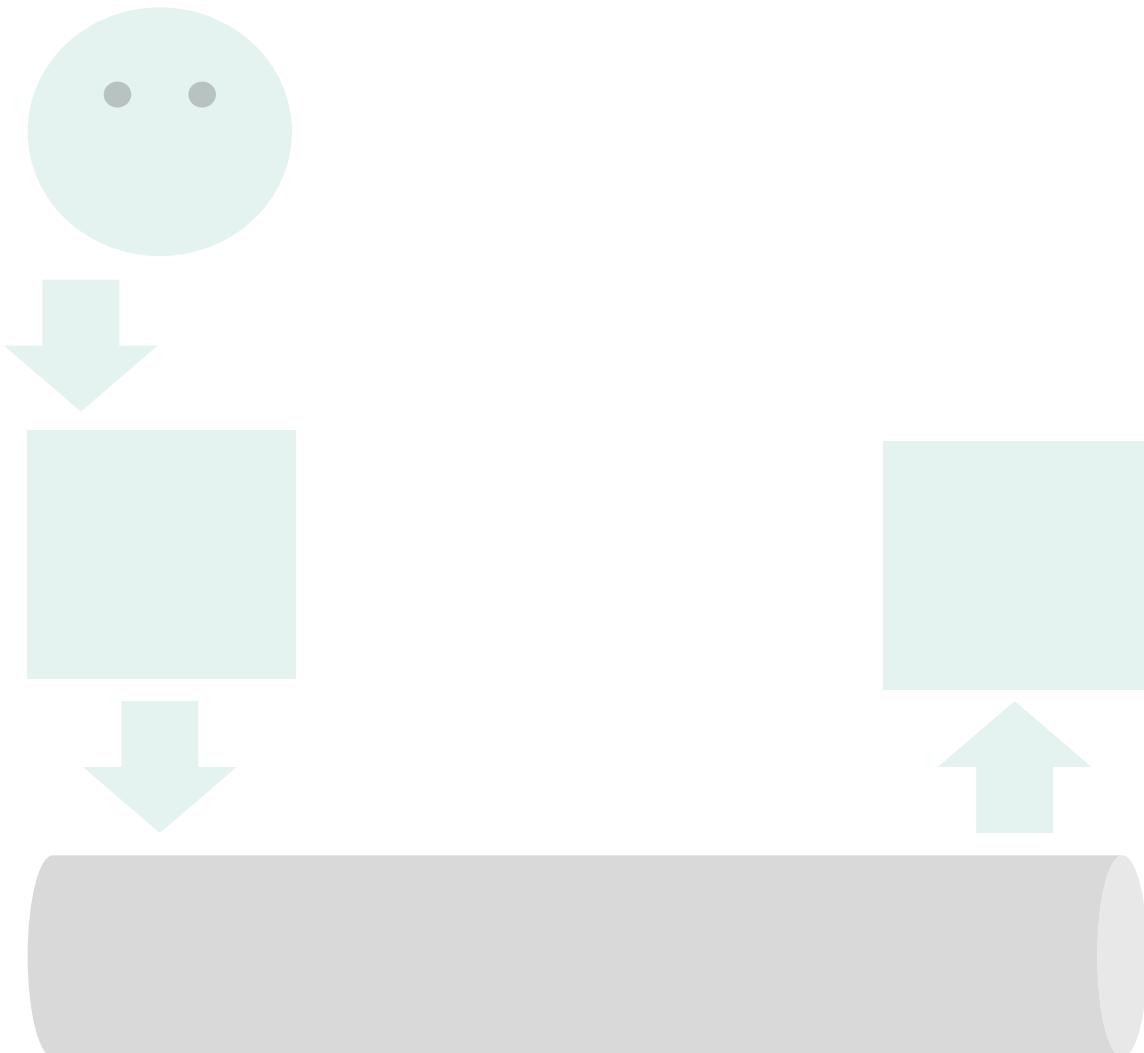
Набросок части архитектуры



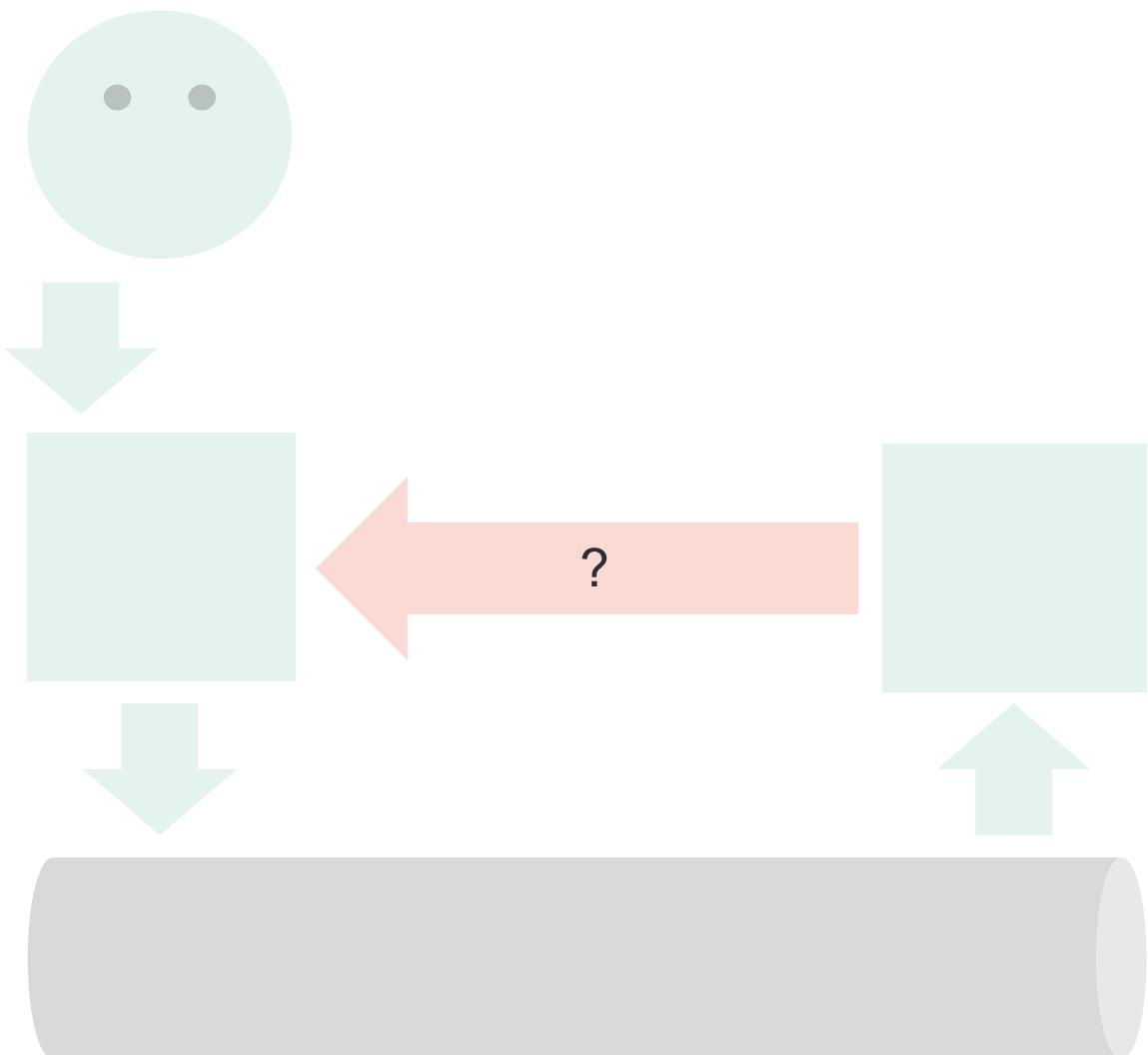
Набросок части архитектуры



Набросок части архитектуры

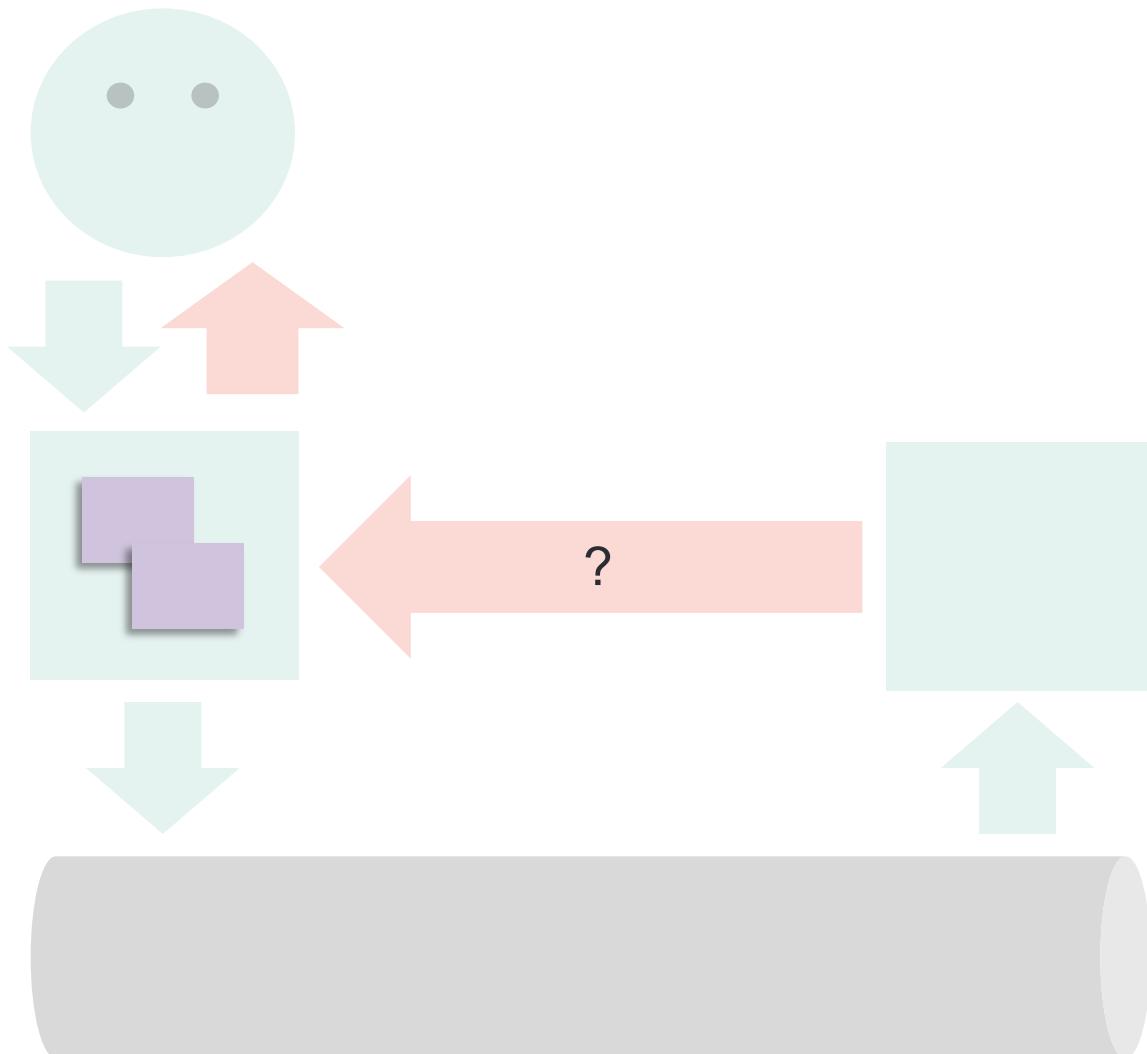


Набросок части архитектуры



Набросок части архитектуры

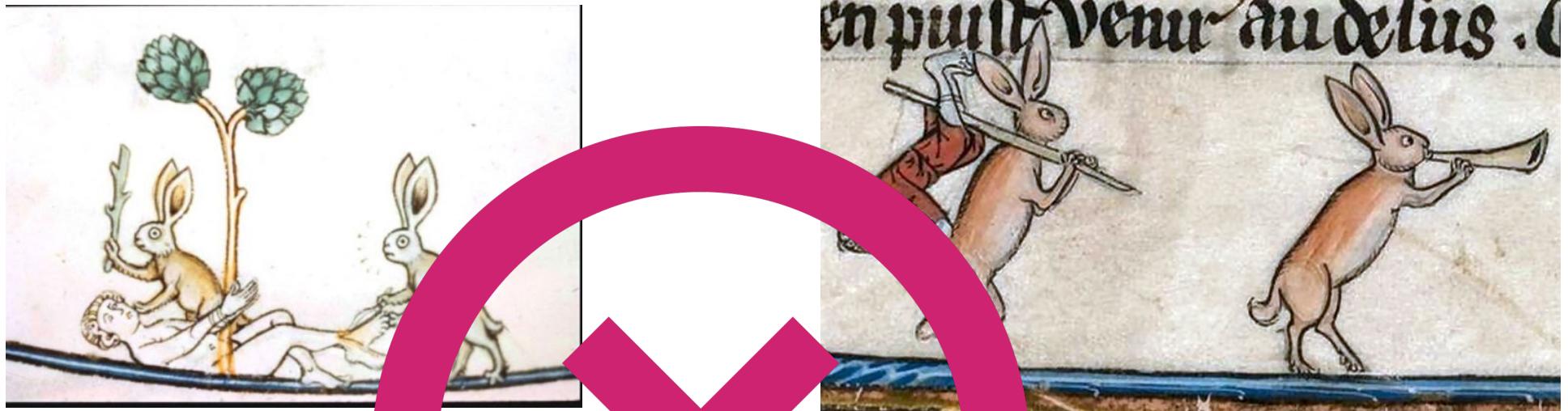
Вот и сложность канала «вверх»



Как же быть с каналом «вверх»?

Например, взять RabbitMQ. Но нет уверенности, что:

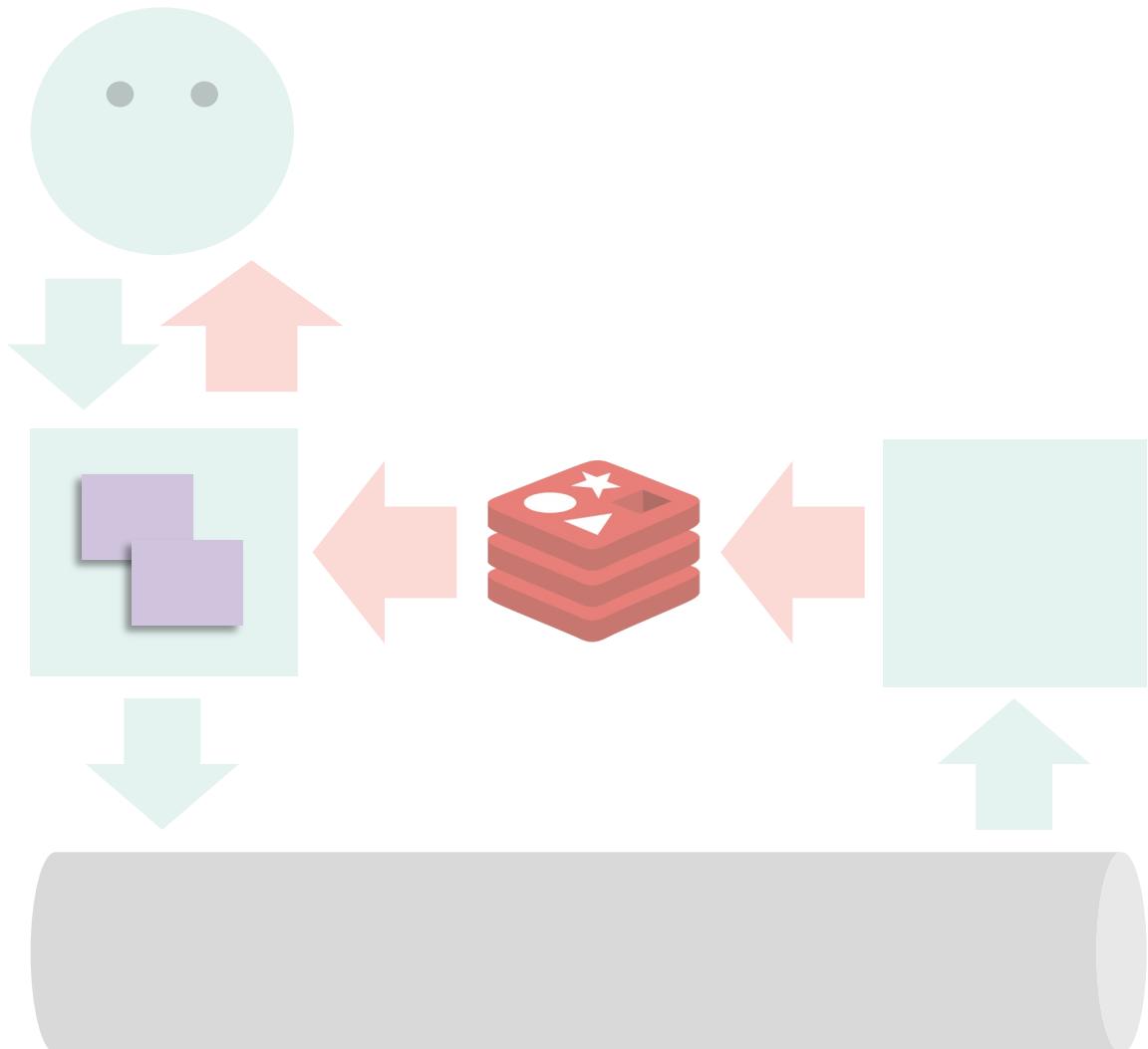
- динамические подписки/отписки быстро работают
- динамическое создание очередей быстро работает
- что это все держит высокую нагрузку



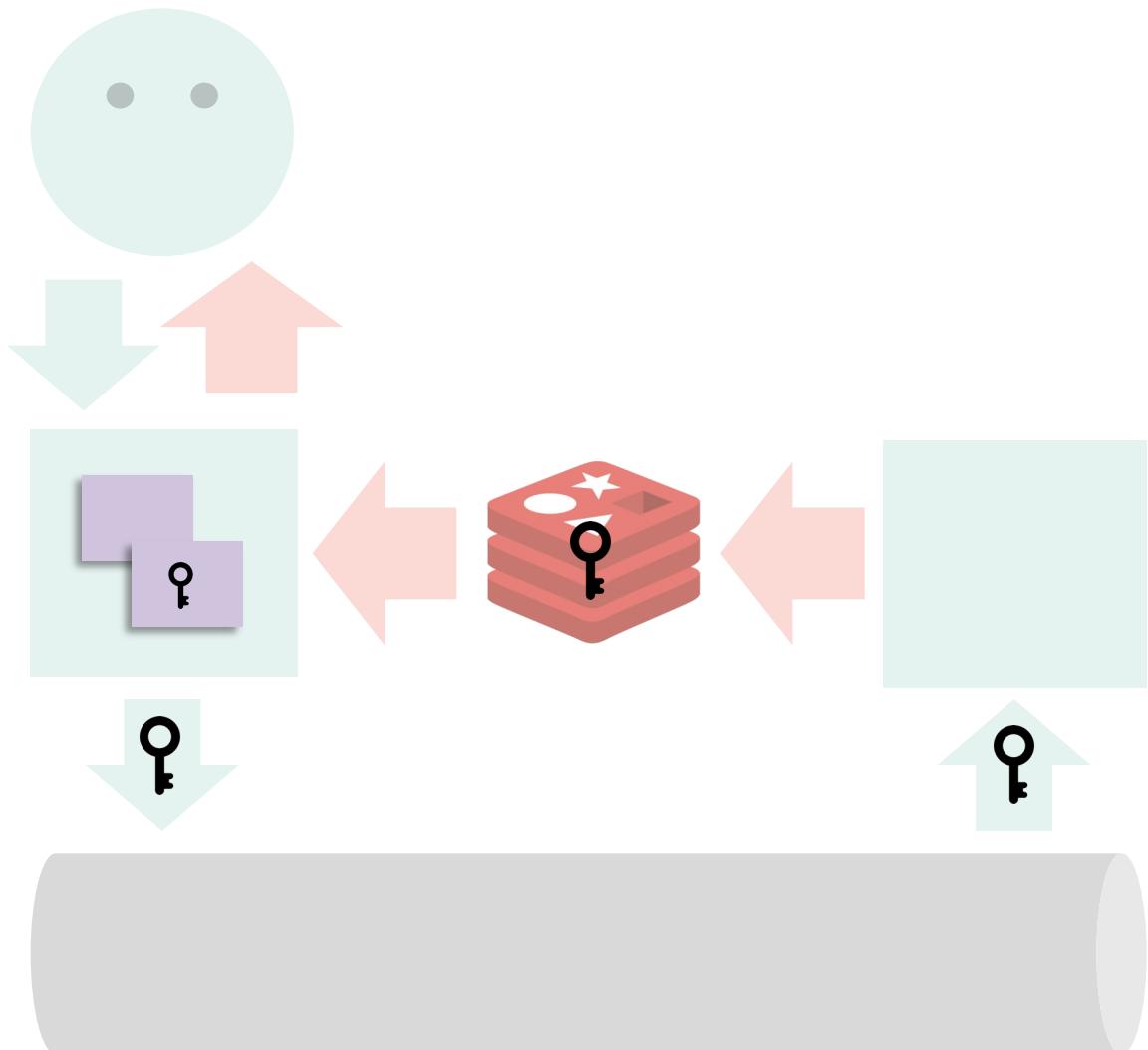
Как же мы решили «проблему 1.0»?



Приблизительно так



Приблизительно так





Как устроен «фронт»

```
WAIT_TIMEOUT: Final[int] = 300
```

```
async def bot_request_handler():
    #...
    inner_key: str = uuid4.uuid() # <-- вот он наш ключ
    await rabbitmq_connection.publish_message(
        'sometopic', {'return_key': inner_key, 'message': 'Сообщение от пользователя'}
    )
    #...
    while True:
        possible_result: bytes | None = await redis_connection.get(inner_key) # <-- вот он наш ключ
        if possible_result:
            await redis_connection.remove(inner_key)
            return JsonResponse({'answer': possible_result})
        await asyncio.sleep(WAIT_TIMEOUT)
    # ... что-то про таймауты
```

Как устроен «бэк» (ml ядро)

MESSAGE_EXPIRATION: Final[int] = 600

```
async def ml_core_wannabe_answerer():
    ...
    rabbit_listener: RabbitmqListener = rabbitmq_connection.subscribe('sometopic')
    while True:
        new_message: dict = await rabbit_listener.read_json()
        ...
        result_of_predict: dict = do_some_predict_magic(new_message)
        ...
        redis_connection.set(
            new_message['return_key'], # <-- ВОТ ОН наш ключ
            json.encode(result_of_predict), MESSAGE_EXPIRATION)
```

Выводы по решению 1.0

- ✓ ну... оно работает!
- ✓ сделано просто и понятно
- ✓ бизнес очень доволен

Выводы по решению 1.0

- ✓ ну... оно работает!
- ✓ сделано просто и понятно
- ✓ бизнес очень доволен
- ✗ цикл со sleep
- ✗ редис неудобно масштабируется
- ✗ у нас «распределенный монолит»,
где несколько сервисов пользуются
«shared DB»

Этот подход в ряде
случаев имеет право
на существование

Подход к проблеме 2.0

Что за продукт связан с «проблемой 2.0»?

— чат + чат-«админка», b2c

Что за продукт связан с «проблемой 2.0»?

- чат + чат-«админка», b2c
- клиенты с текстовыми обращениями приходят сюда

Что за продукт связан с «проблемой 2.0»?

- чат + чат-«админка», b2c
- клиенты с текстовыми обращениями приходят сюда
- обслуживает мобильное приложение, виджет на сайте, мессенджеры и т.п.

Что за продукт связан с «проблемой 2.0»?

- чат + чат-«админка», b2c
- клиенты с текстовыми обращениями приходят сюда
- обслуживает мобильное приложение, виджет на сайте, мессенджеры и т.п.
- имеет много интеграций, несколько своих фронтендов

Что за продукт связан с «проблемой 2.0»?

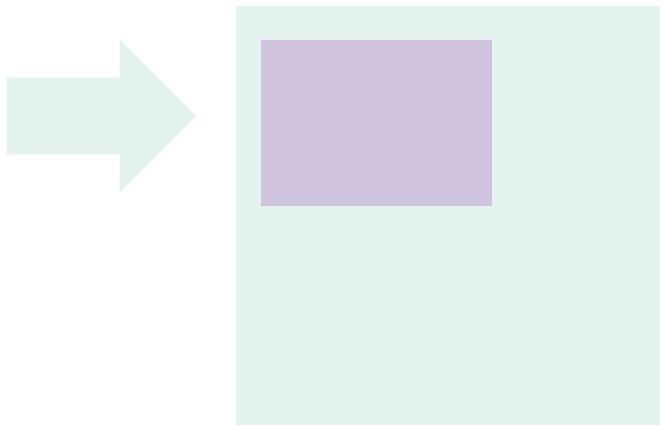
- чат + чат-«админка», b2c
- клиенты с текстовыми обращениями приходят сюда
- обслуживает мобильное приложение, виджет на сайте, мессенджеры и т.п.
- имеет много интеграций, несколько своих фронтендов
- активно разрабатывается, бизнес требует большого объема функциональности

А также

- MSA архитектура («шареный» redis не пойдет)
- Кубер

Общая архитектура

В общих чертах



Общая архитектура

В общих чертах



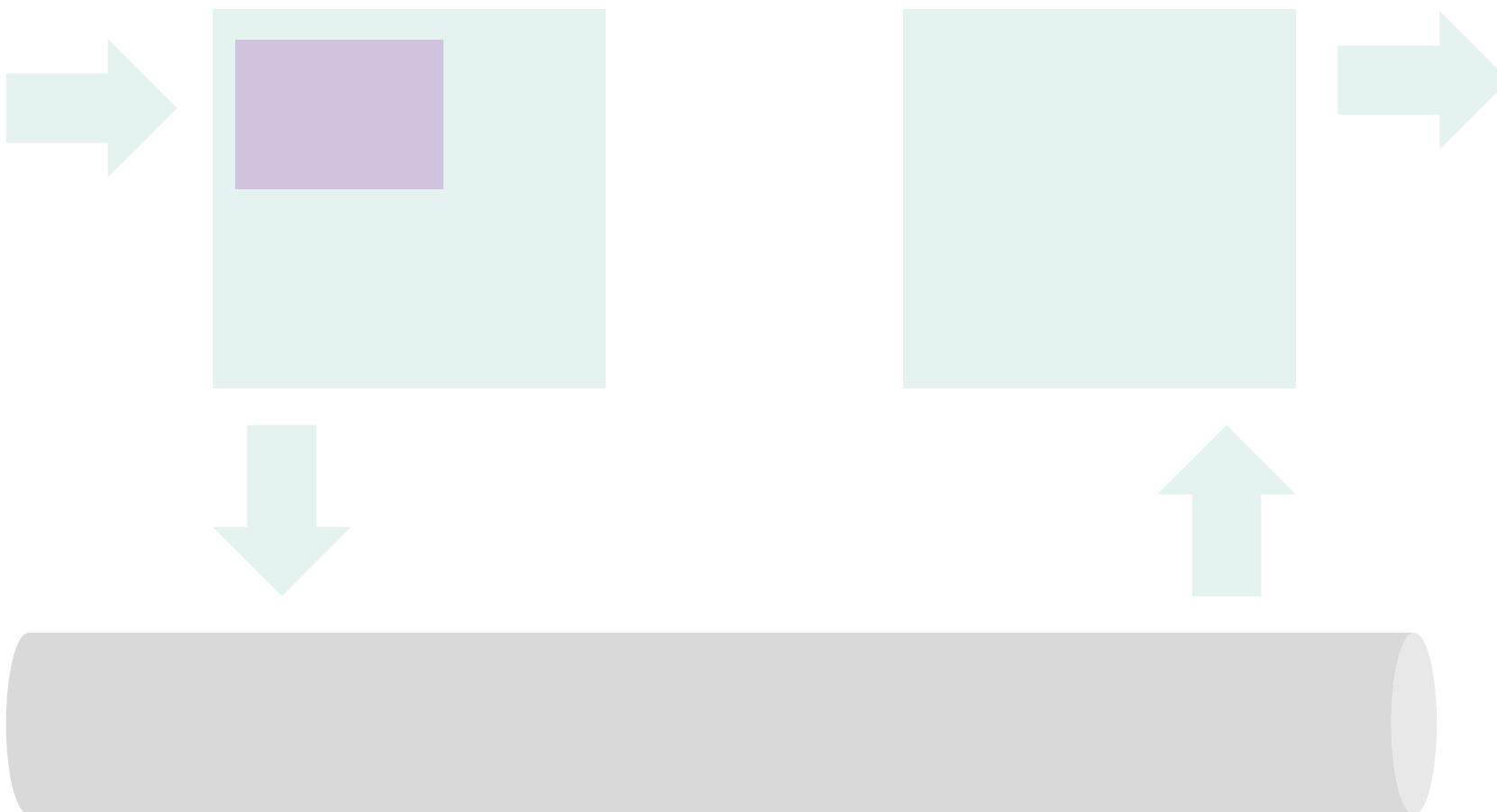
Общая архитектура

В общих чертах



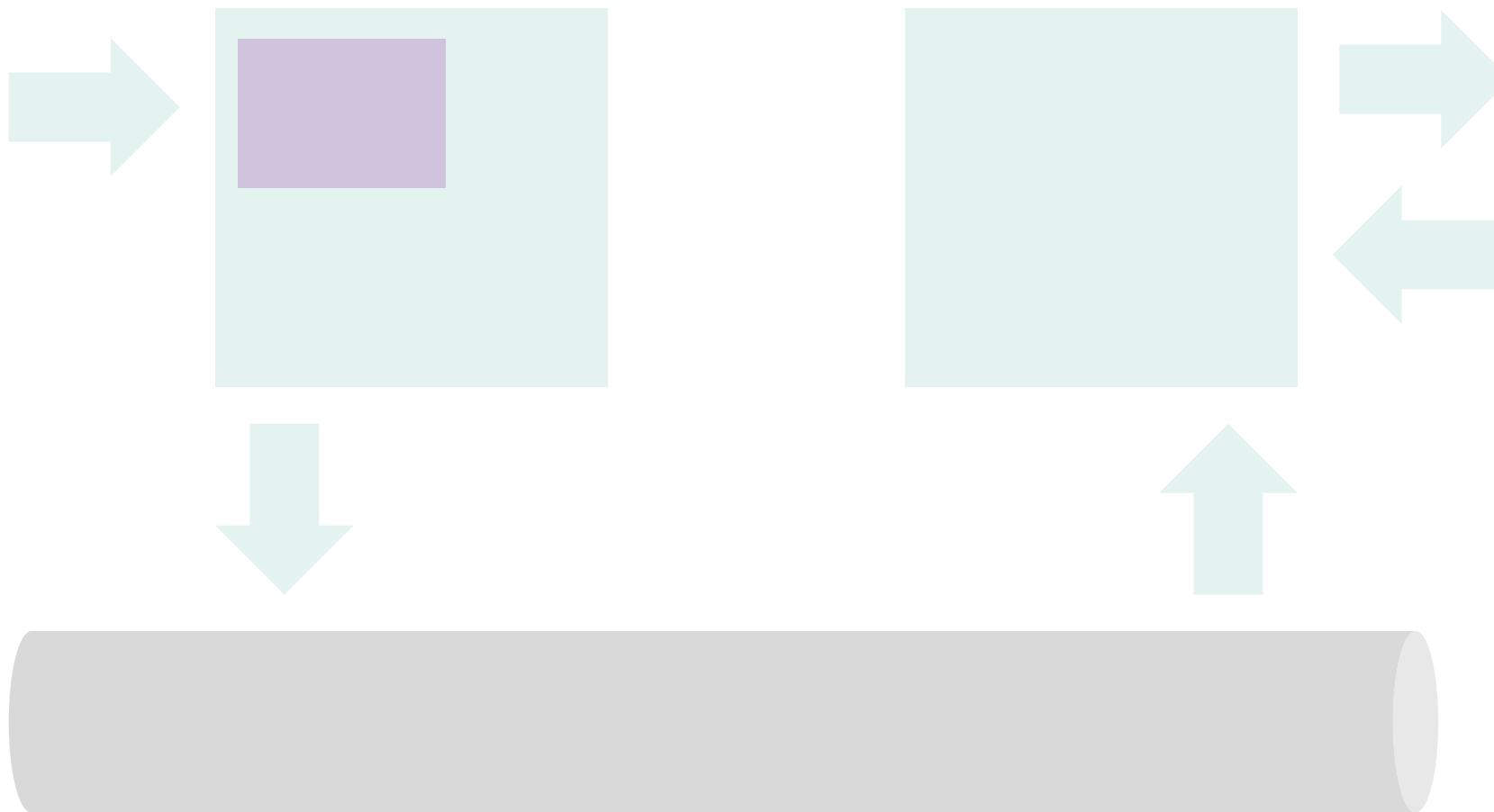
Общая архитектура

В общих чертах



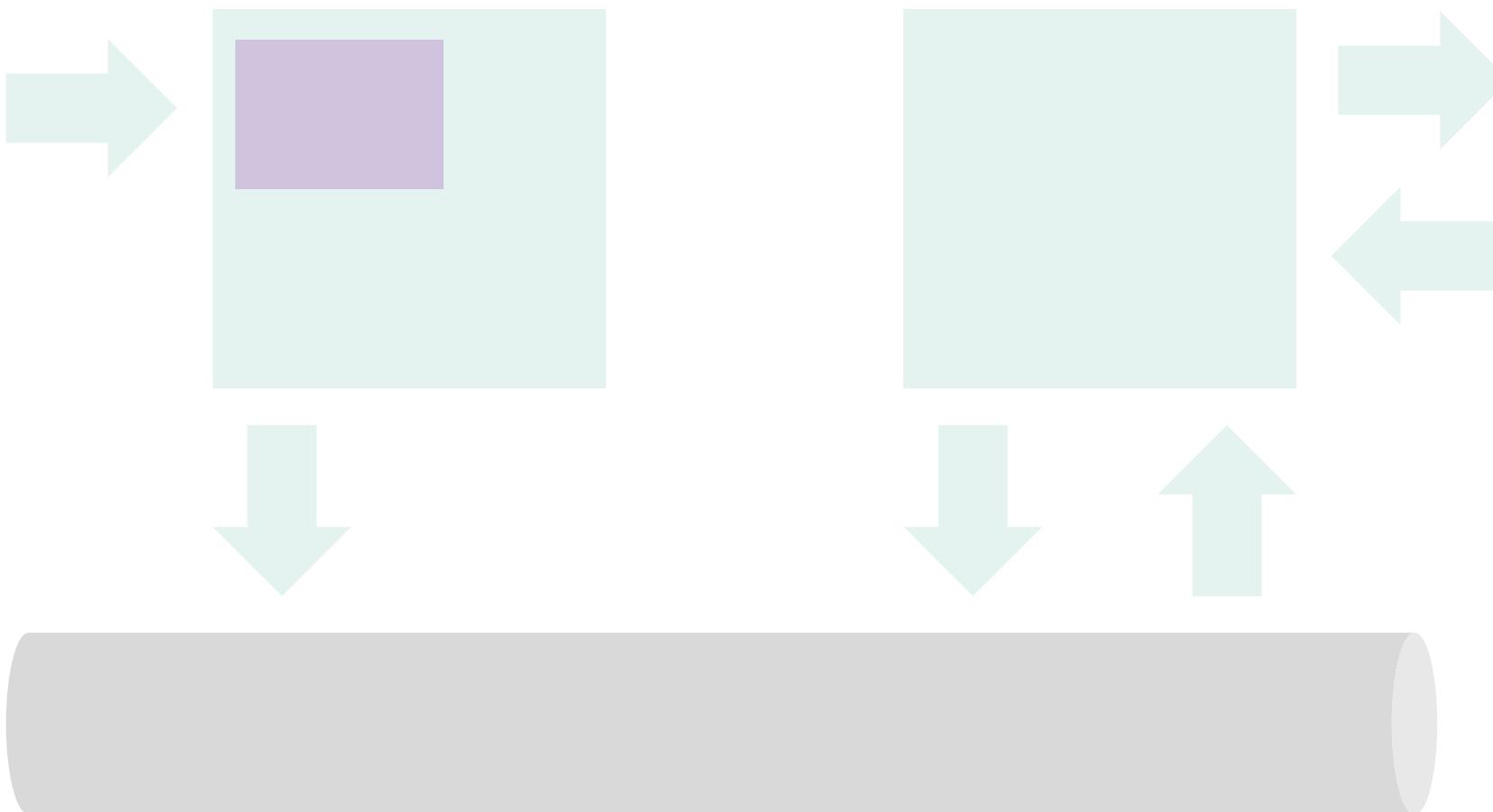
Общая архитектура

В общих чертах



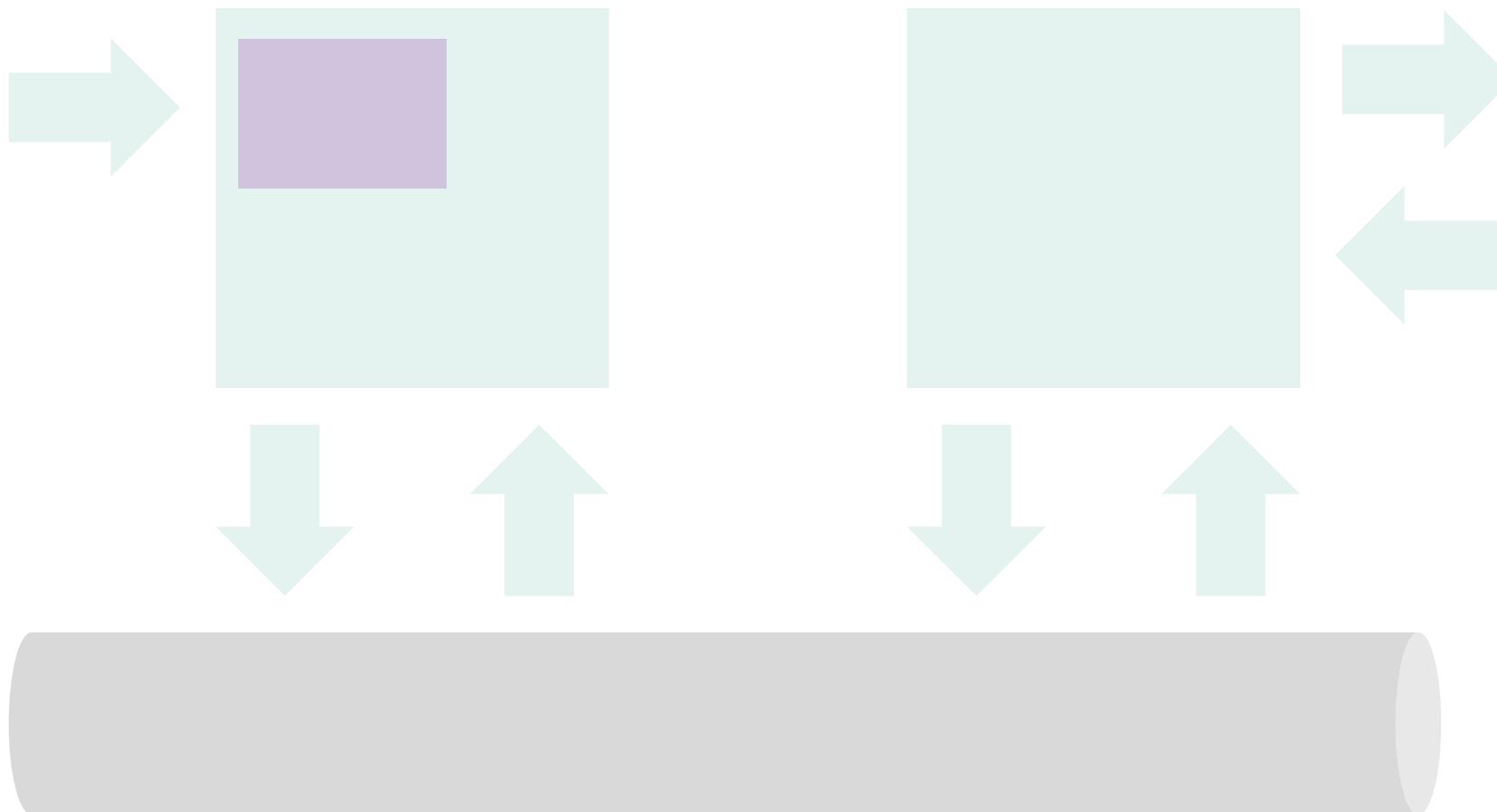
Общая архитектура

В общих чертах



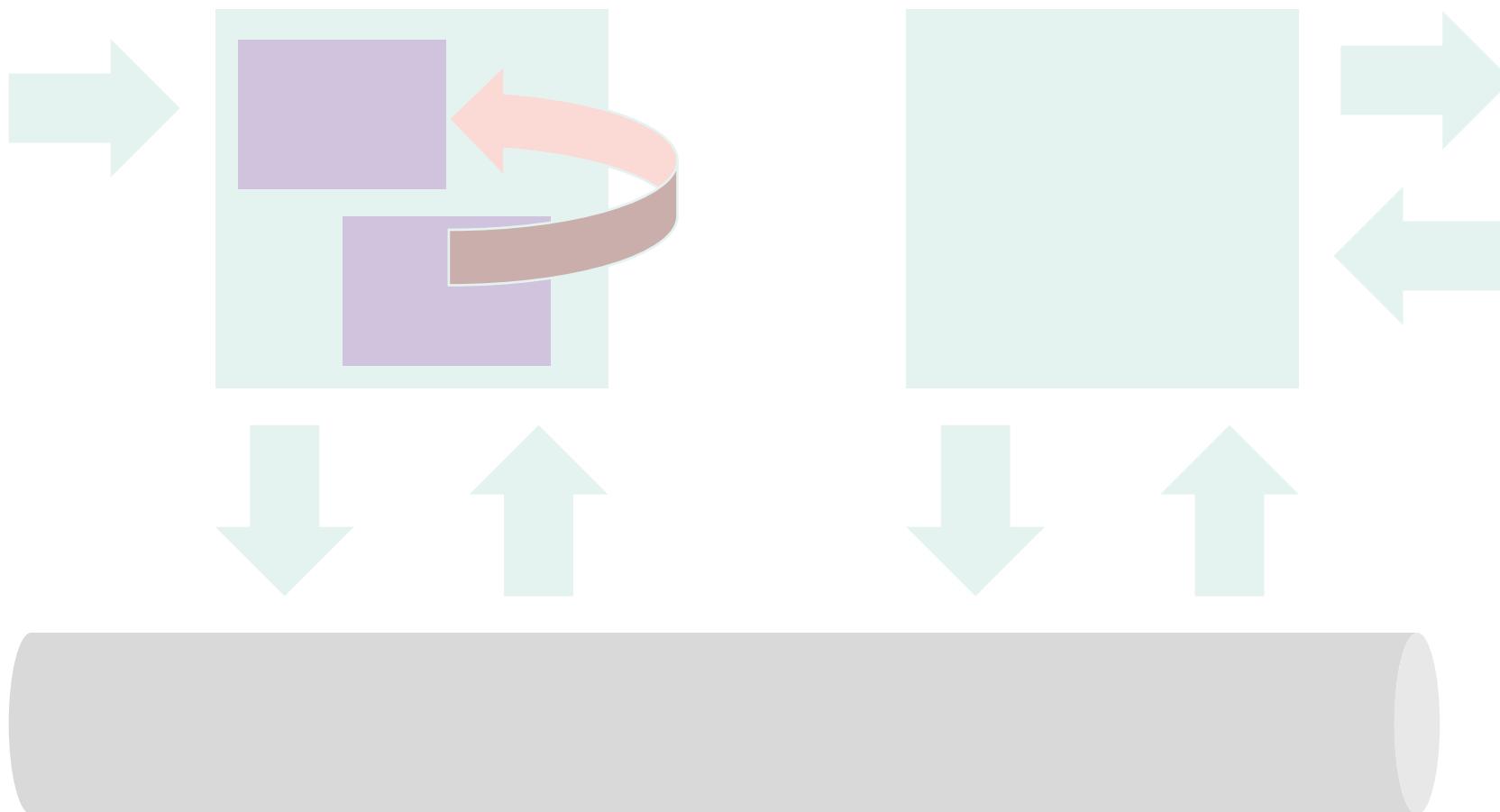
Общая архитектура

В общих чертах



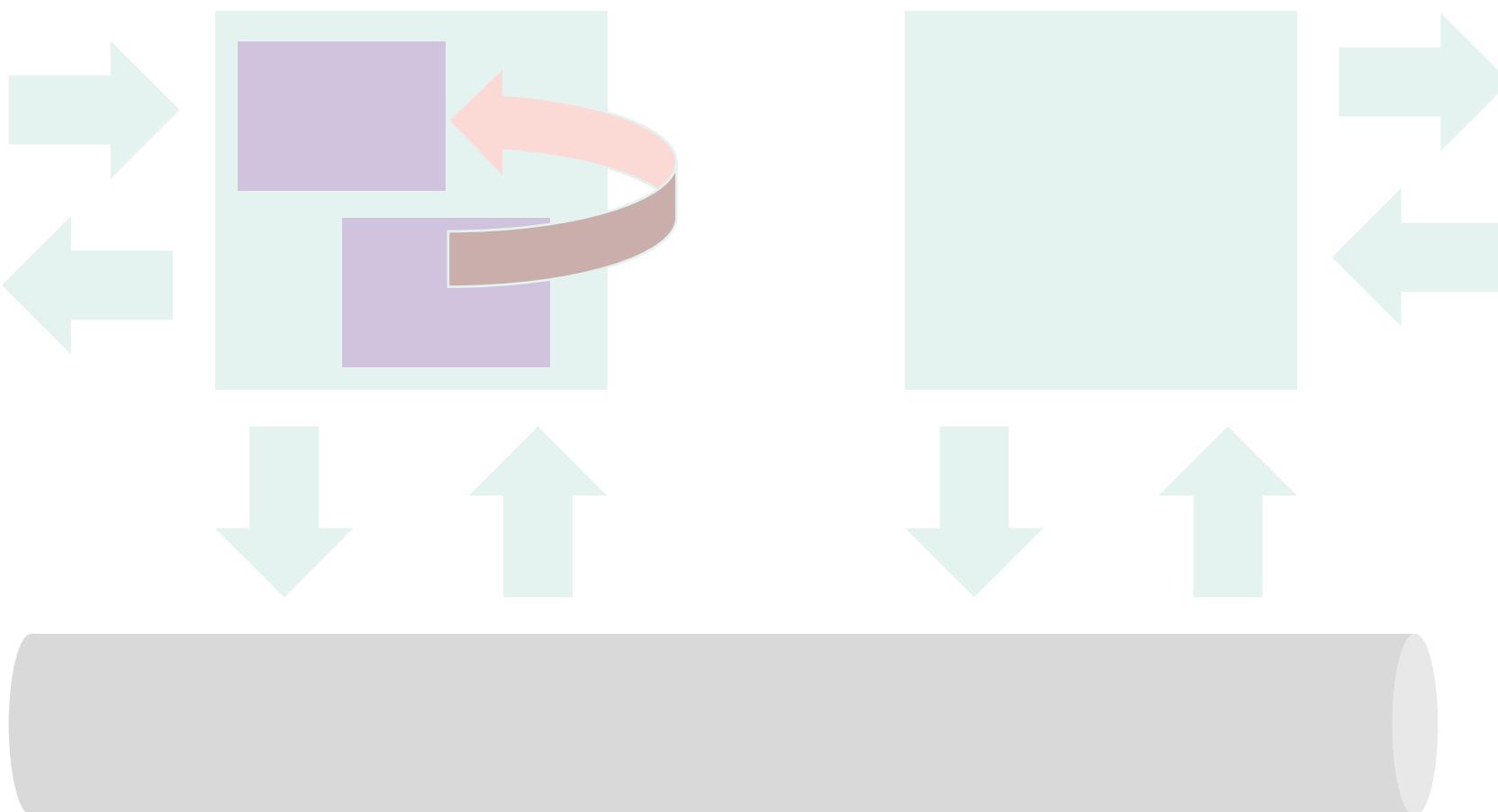
Общая архитектура

В общих чертах



Общая архитектура

В общих чертах



Канал «ВНИЗ»

Нам пришло сообщение от пользователя и мы...

Юзеры пишут

JWT

Валидируем JWT

Валидируем структуры данных,
формируем выходную структуру и
сериализуем в JSON

Отправляем в Kafka

Канал «ВНИЗ»

Нам пришло сообщение от пользователя и мы...

Валидируем JWT

Валидируем структуры данных,
формируем выходную структуру и
сериализуем в JSON

Отправляем в Kafka

Юзеры пишут

JWT

Pydantic
валидация

Канал «ВНИЗ»

Нам пришло сообщение от пользователя и мы...

Валидируем JWT

Валидируем структуры данных,
формируем выходную структуру и
сериализуем в JSON

Отправляем в Kafka

Юзеры пишут

JWT

Pydantic
валидация

Pydantic
формируем
выходную ст.

Канал «ВНИЗ»

Нам пришло сообщение от пользователя и мы...

Валидируем JWT

Валидируем структуры данных,
формируем выходную структуру и
сериализуем в JSON

Отправляем в Kafka

Юзеры пишут

JWT

Pydantic
валидация

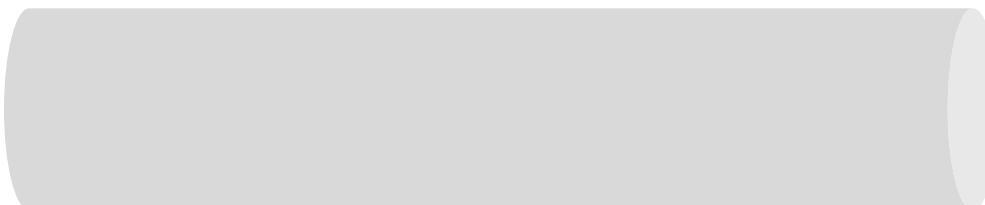
Pydantic
формируем
выходную ст.

Kafka

Всё довольно просто,
но дальше
интереснее...

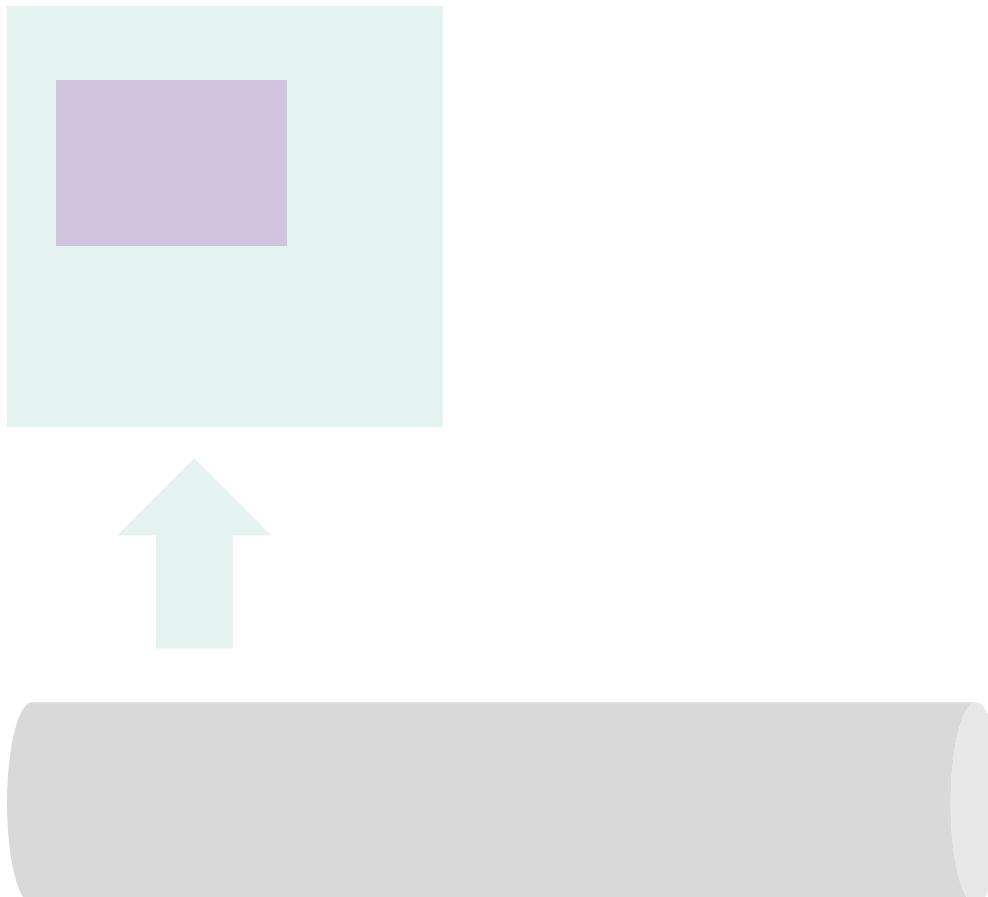
Основная проблема канала «вверх»

Сообщение приходит из недр системы (от оператора, например) и идет к пользователю



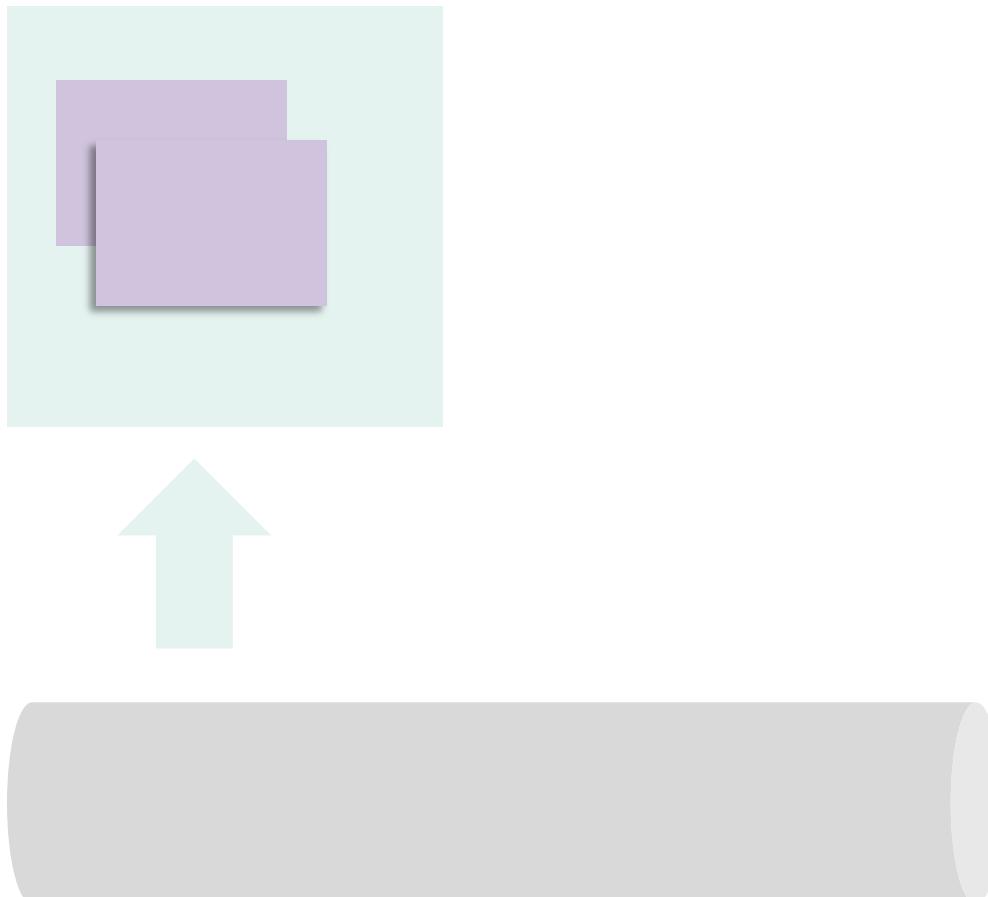
Основная проблема канала «вверх»

Сообщение приходит из недр системы (от оператора, например) и идет к пользователю



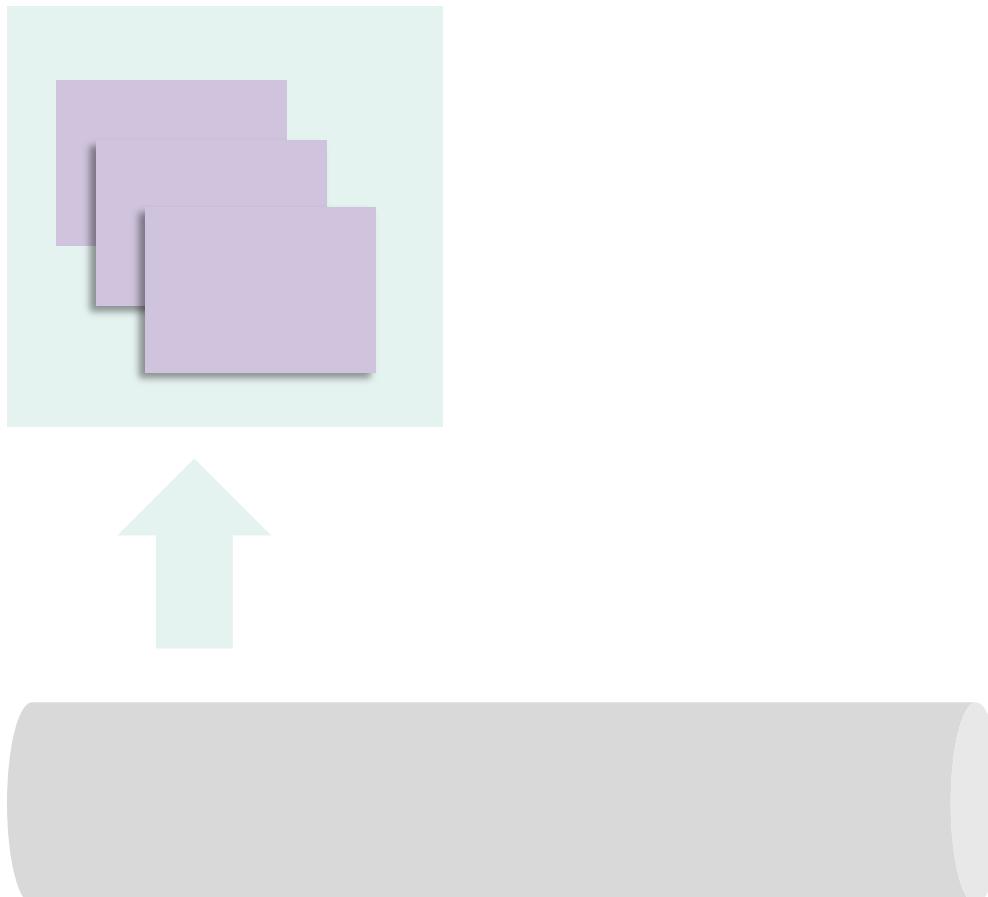
Основная проблема канала «вверх»

Сообщение приходит из недр системы (от оператора, например) и идет к пользователю



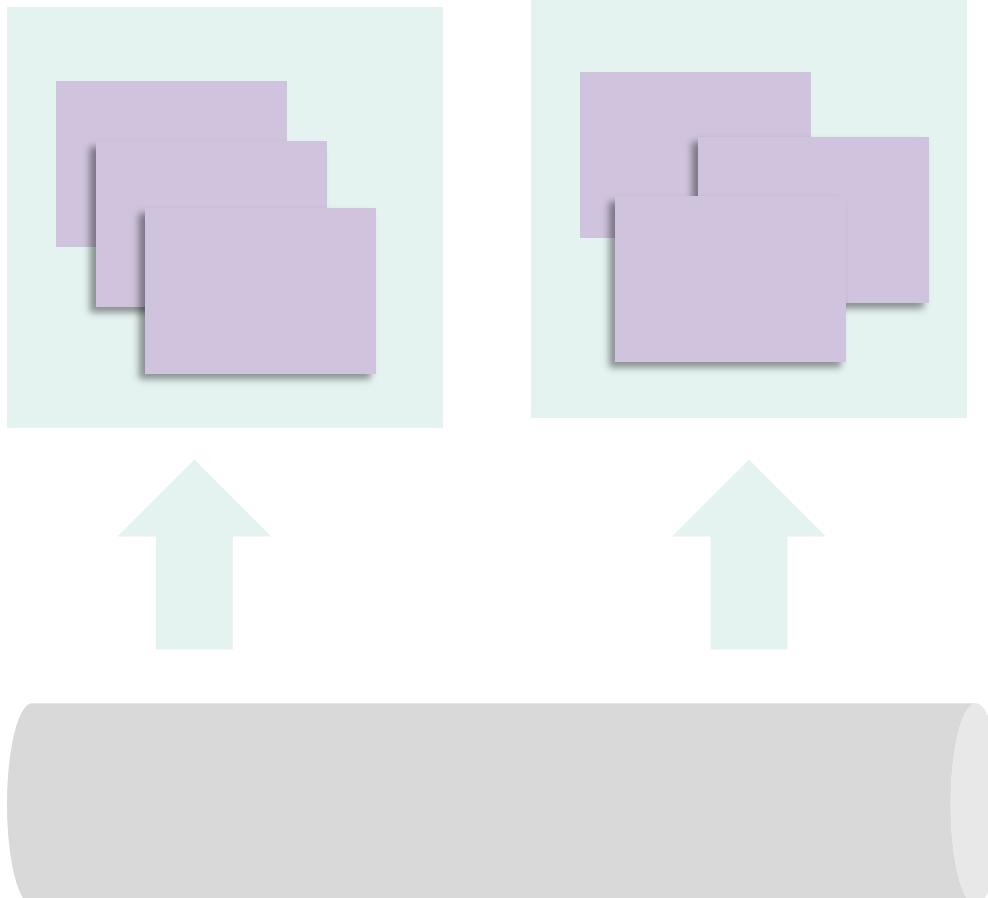
Основная проблема канала «вверх»

Сообщение приходит из недр системы (от оператора, например) и идет к пользователю



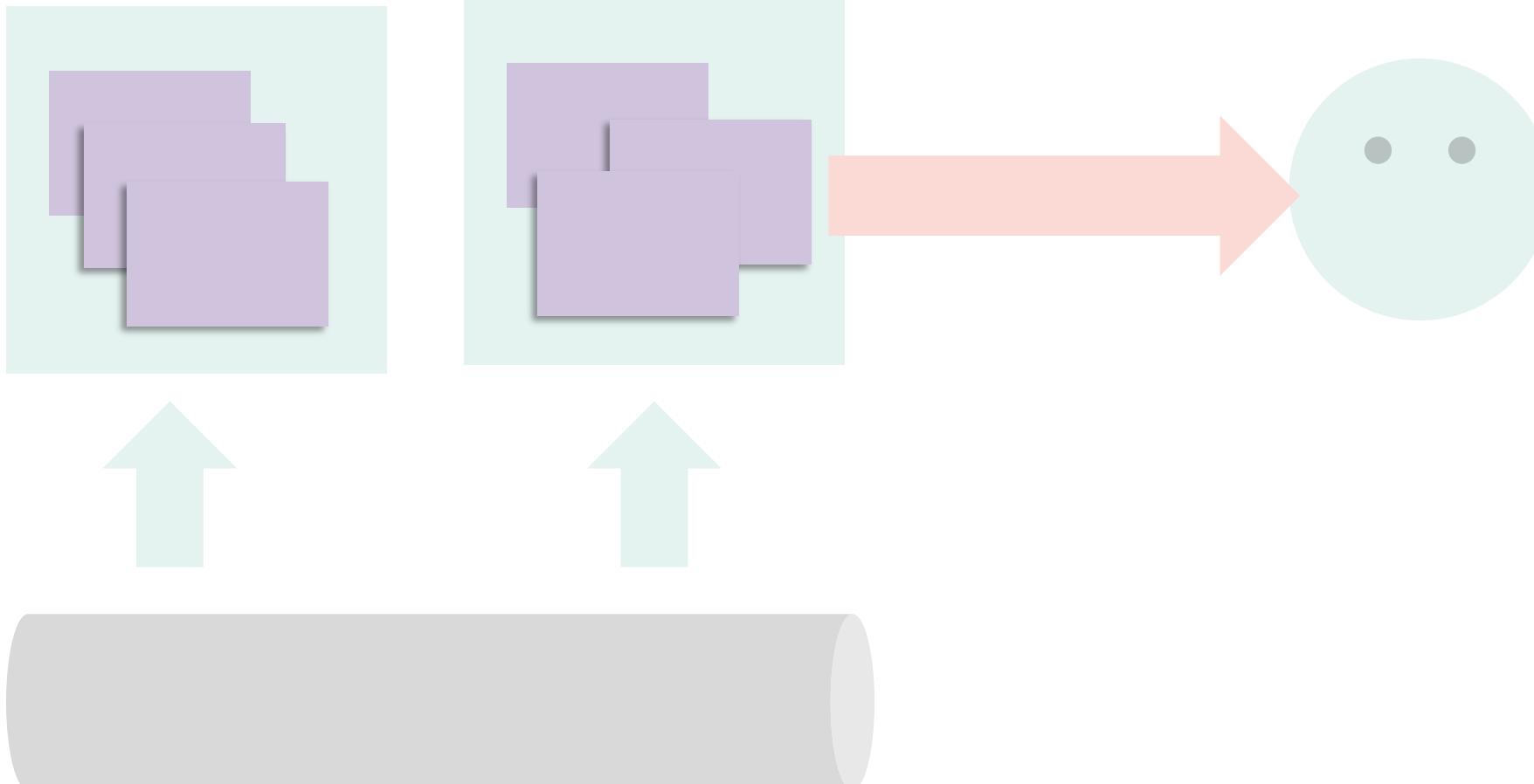
Основная проблема канала «вверх»

Сообщение приходит из недр системы (от оператора, например) и идет к пользователю



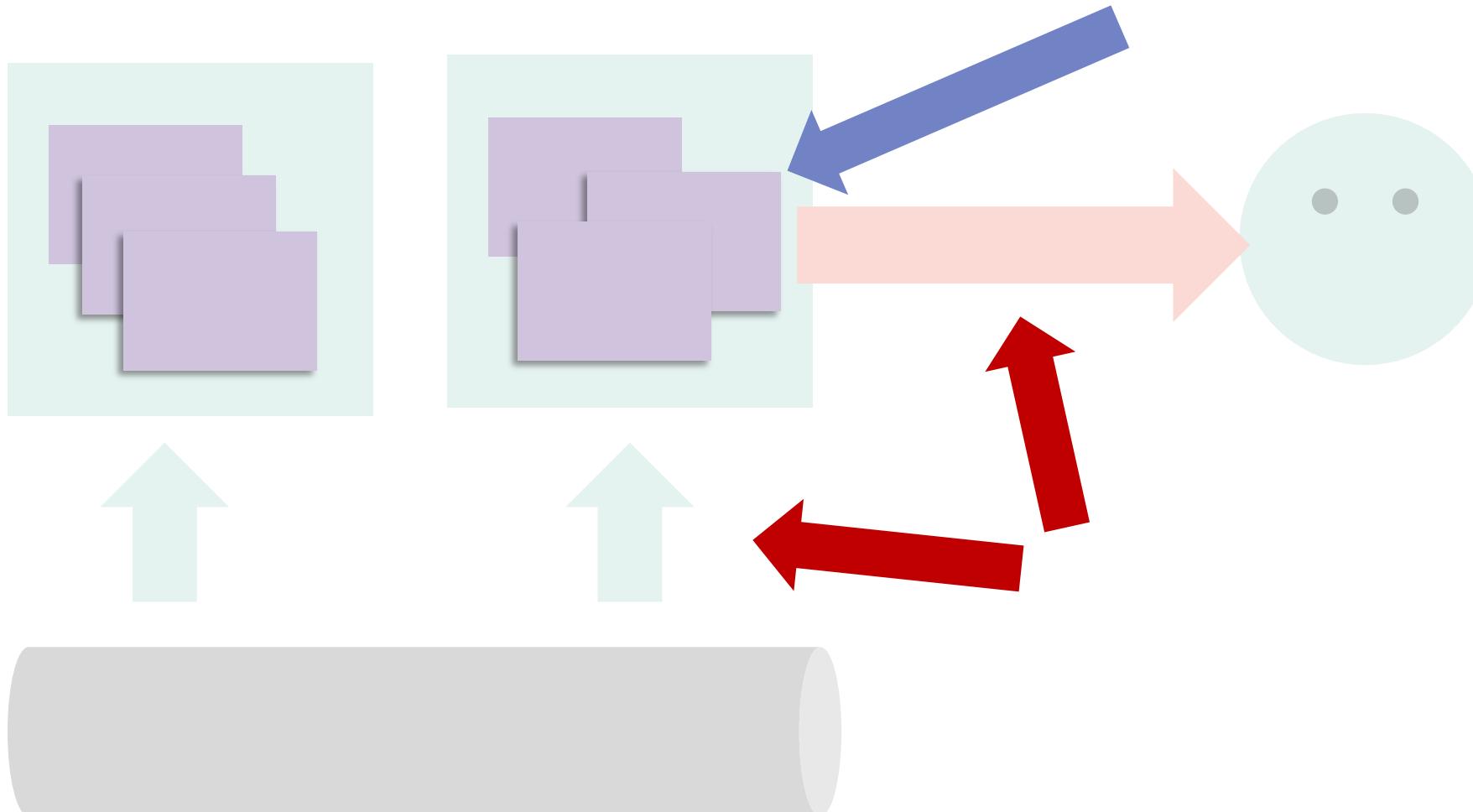
Основная проблема канала «вверх»

Сообщение приходит из недр системы (от оператора, например) и идет к пользователю



Основная проблема канала «вверх»

Сообщение приходит из недр системы (от оператора, например) и идет к пользователю



Как мы решали вопросы

- 1 Написали свой сервис
- 2 Разделили на две части
- 3 Первая часть держит websocket соединение
- 4 Вторая роутит сообщения из кафки, «вверх»
- 5 Части сервиса общаются через ZeroMQ

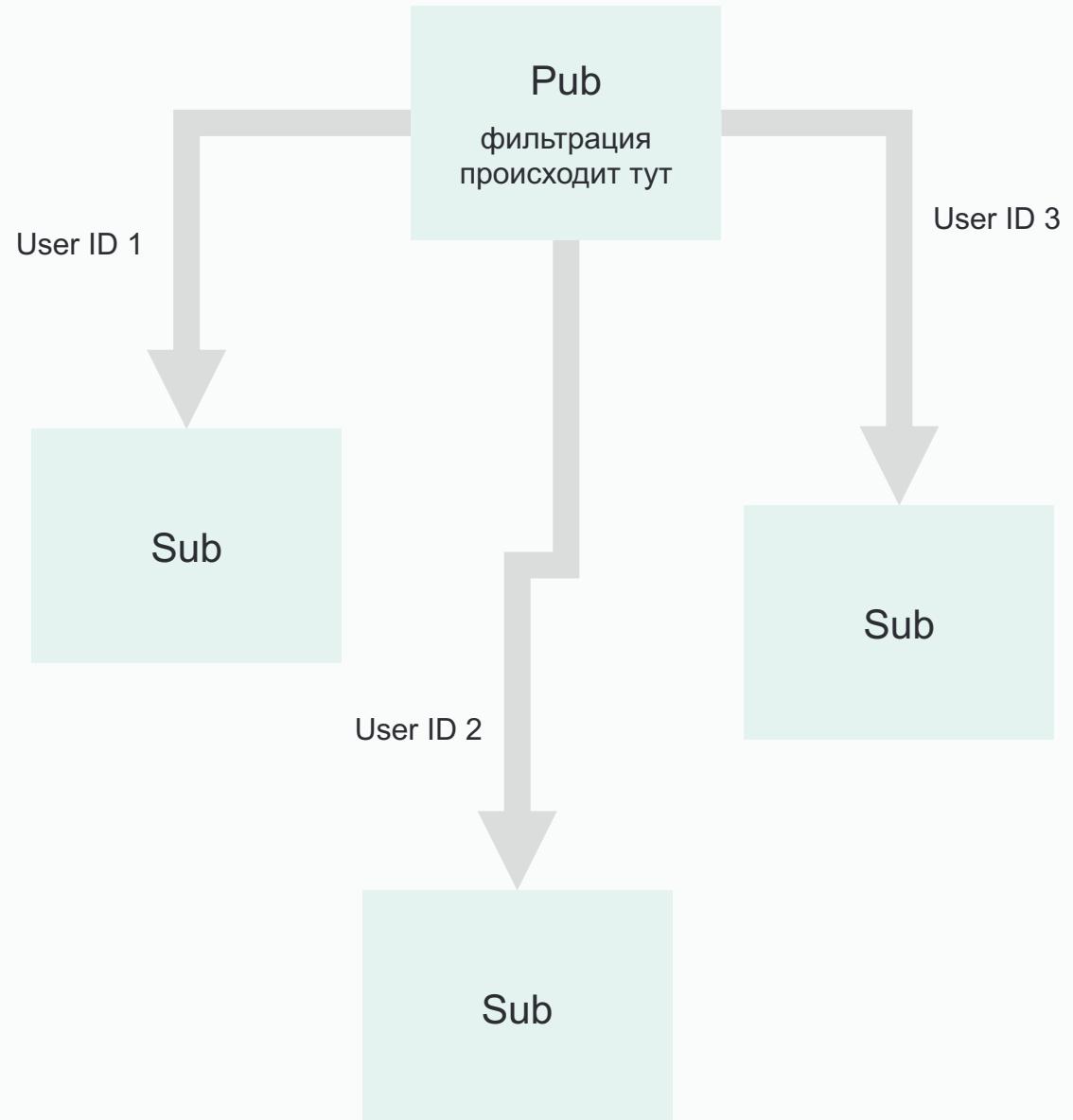
Как назвали части:

- 1 — websocket сервер
- 2 — роутер

Почему ZeroMQ



Быстро!



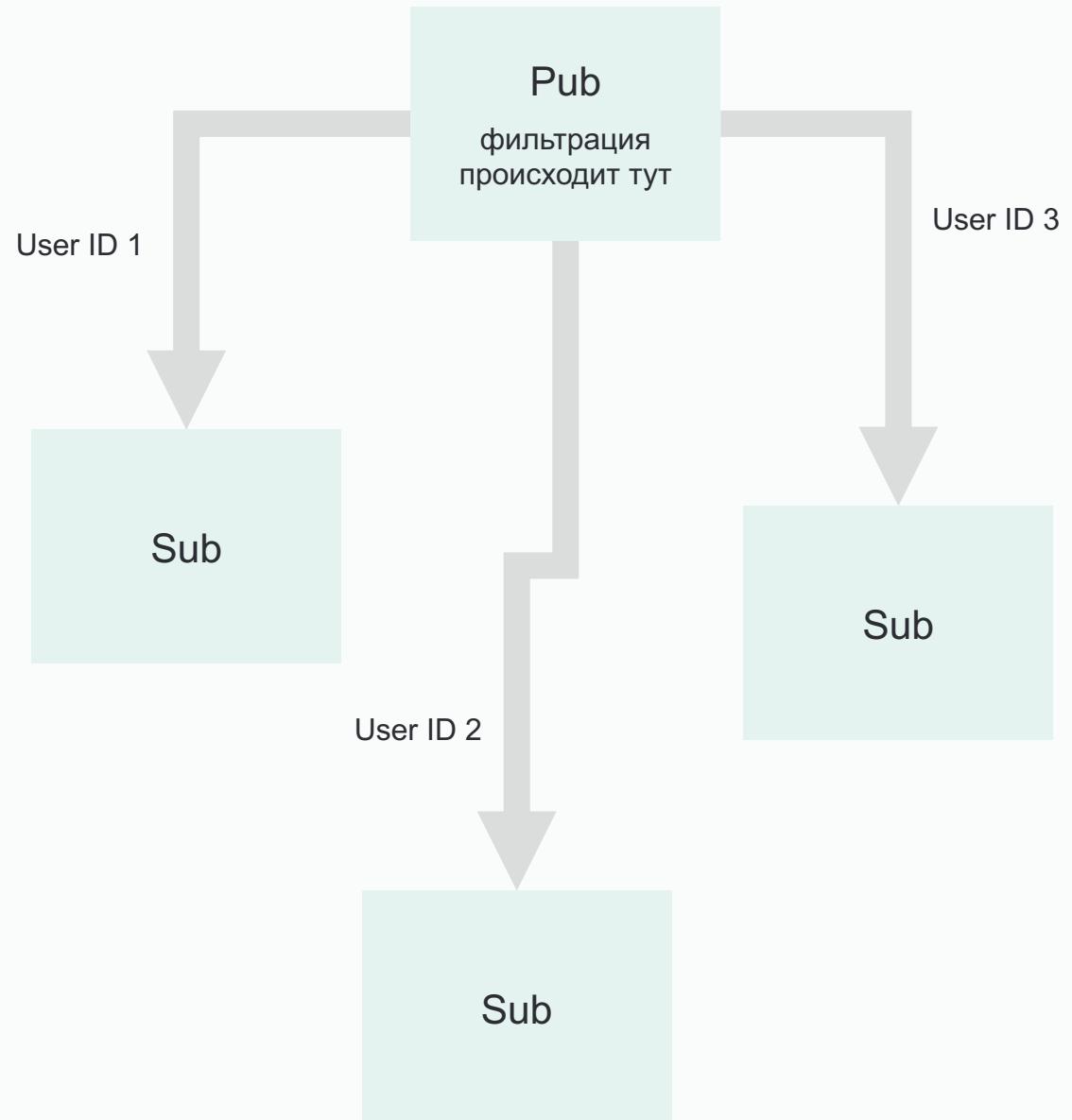
Почему ZeroMQ



Быстро!

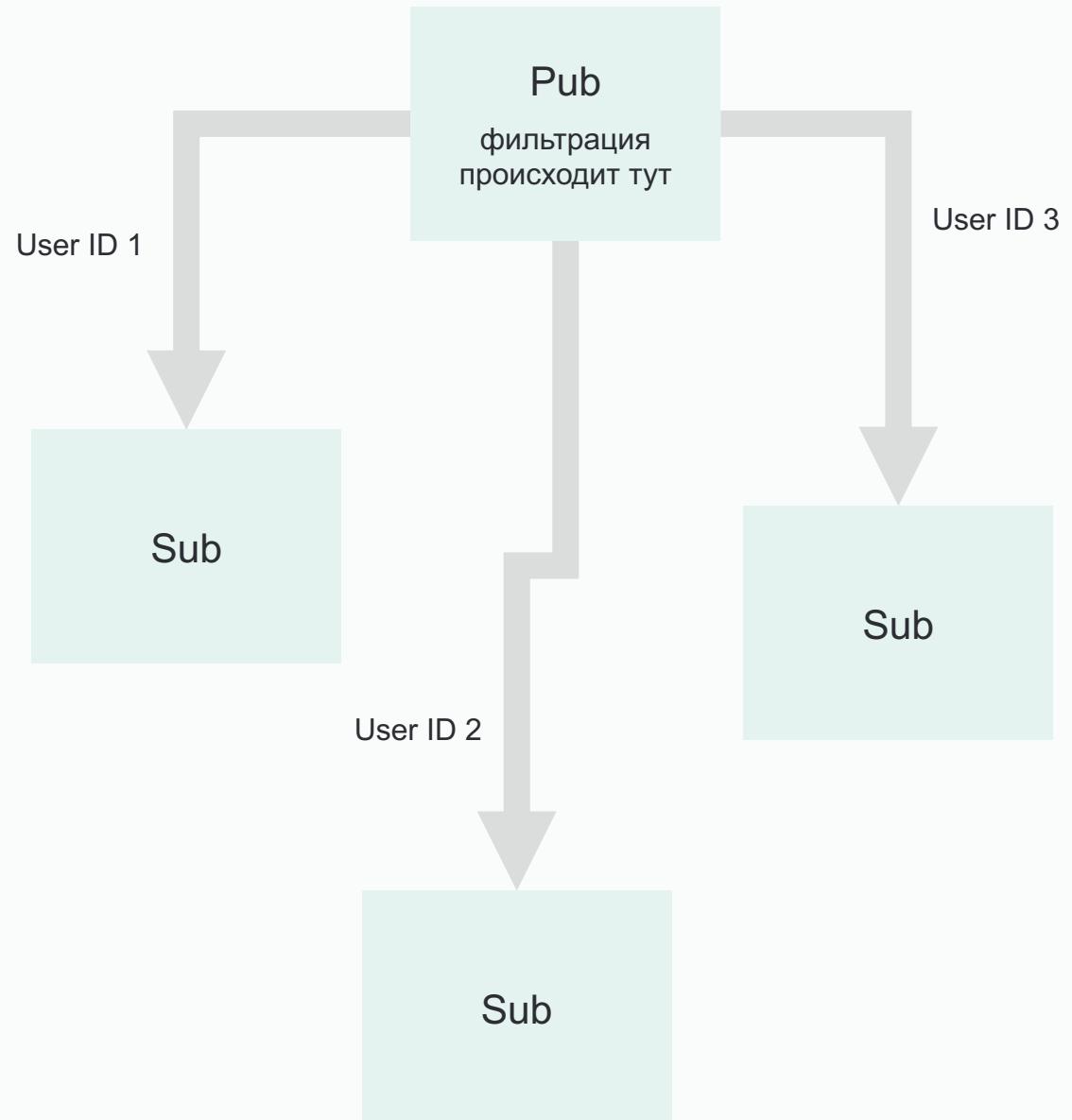


Нет брокера! (мы ленивые)



Почему ZeroMQ

- Быстро!
- Нет брокера! (мы ленивые)
- Есть куча механизмов и биндинги под python



Архитектурная схема роутера

На базе ZeroMQ

— В качестве ключа мы используем ID пользователя

Coroutine 1

в любом поде или
воркере

Router

в любом поде или
воркере

Coroutine 2

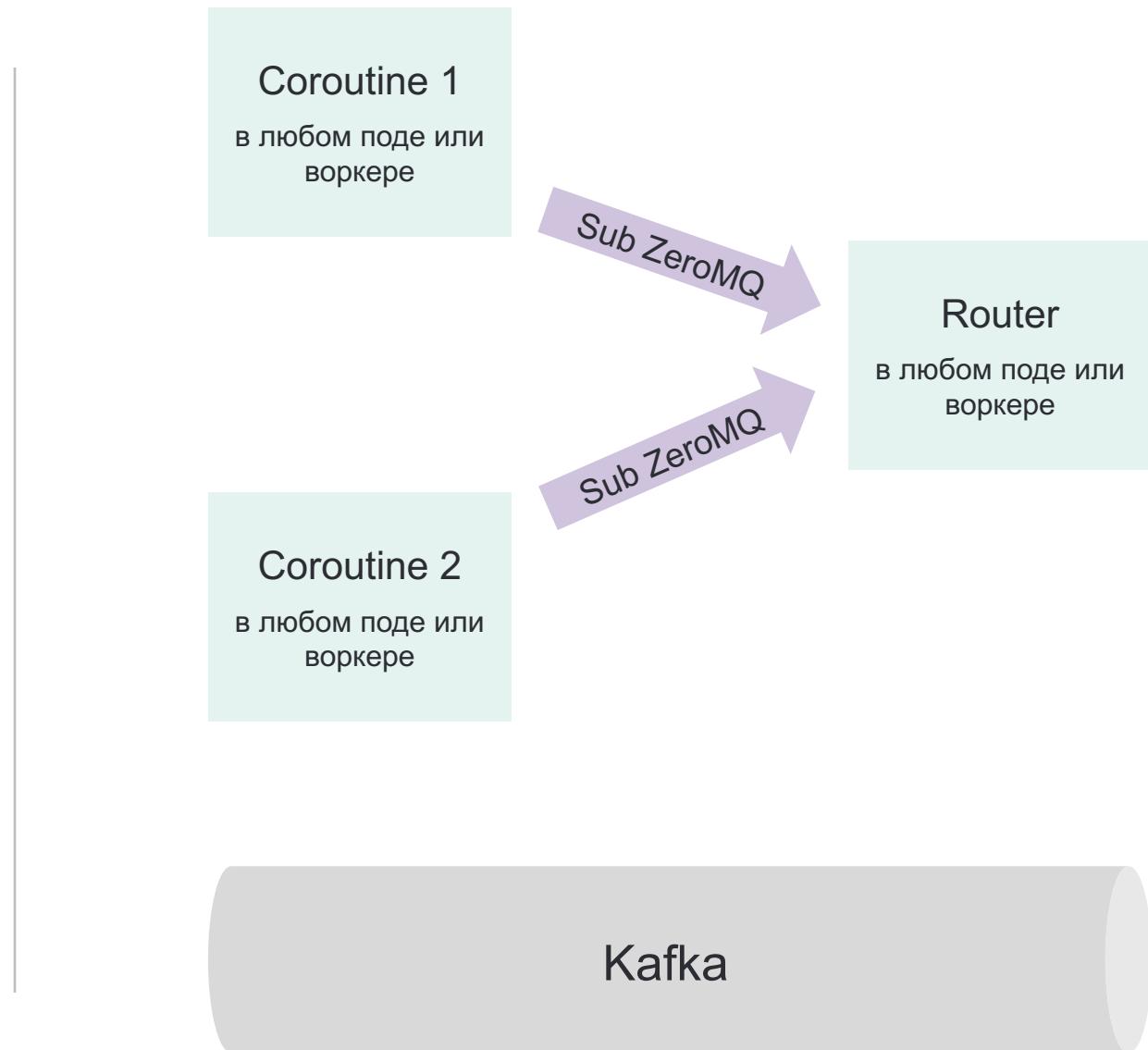
в любом поде или
воркере

Kafka

Архитектурная схема роутера

На базе ZeroMQ

- В качестве ключа мы используем ID пользователя
- Каждая корутина со ссылкой на вебсокет соединение «сабается» на наш роутер



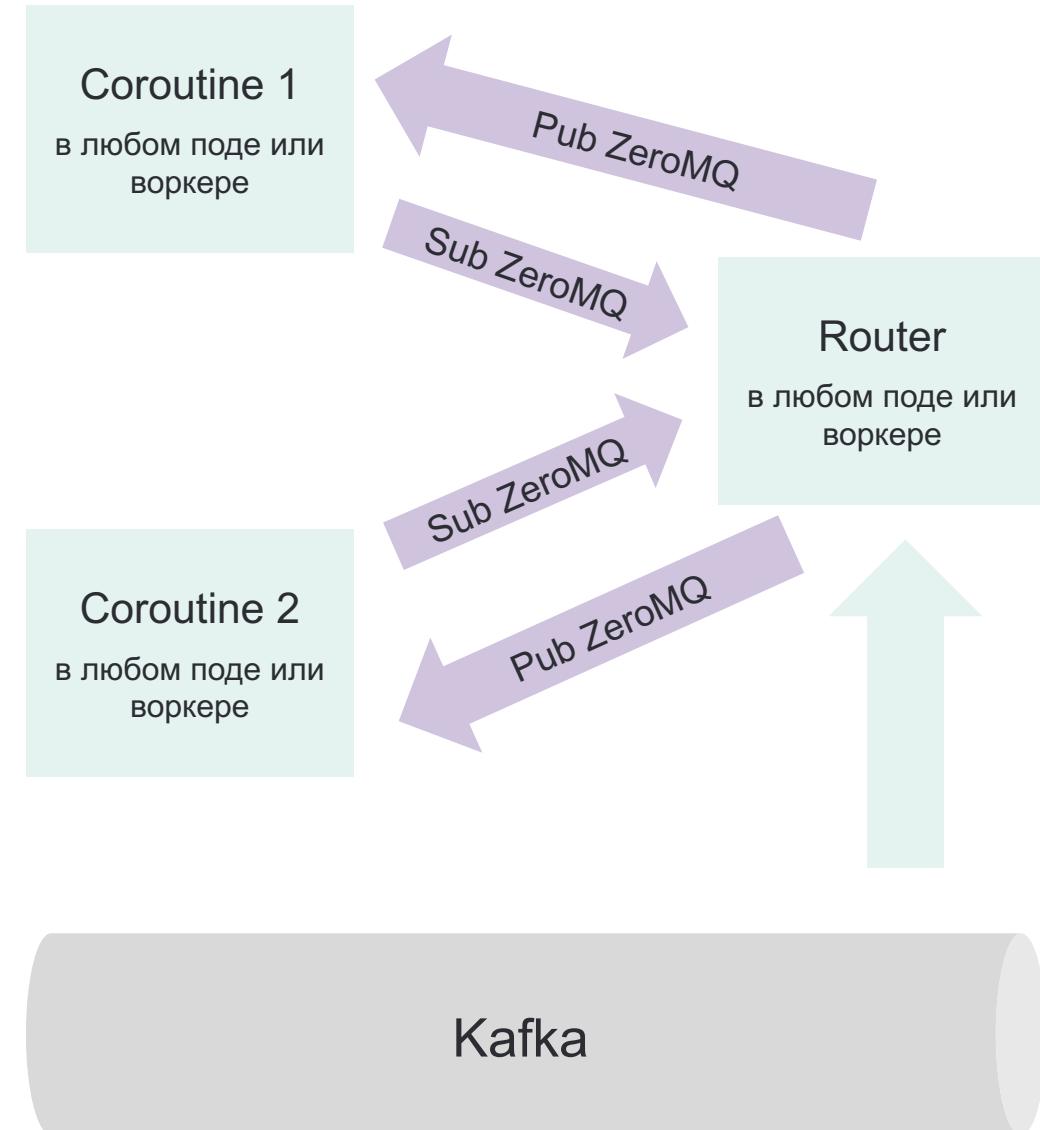
Архитектурная схема роутера

На базе ZeroMQ

- В качестве ключа мы используем ID пользователя
- Каждая корутина со ссылкой на вебсокет соединение «сабается» на наш роутер
- Роутер паблишит «корутинам» по ключу с ID пользователя

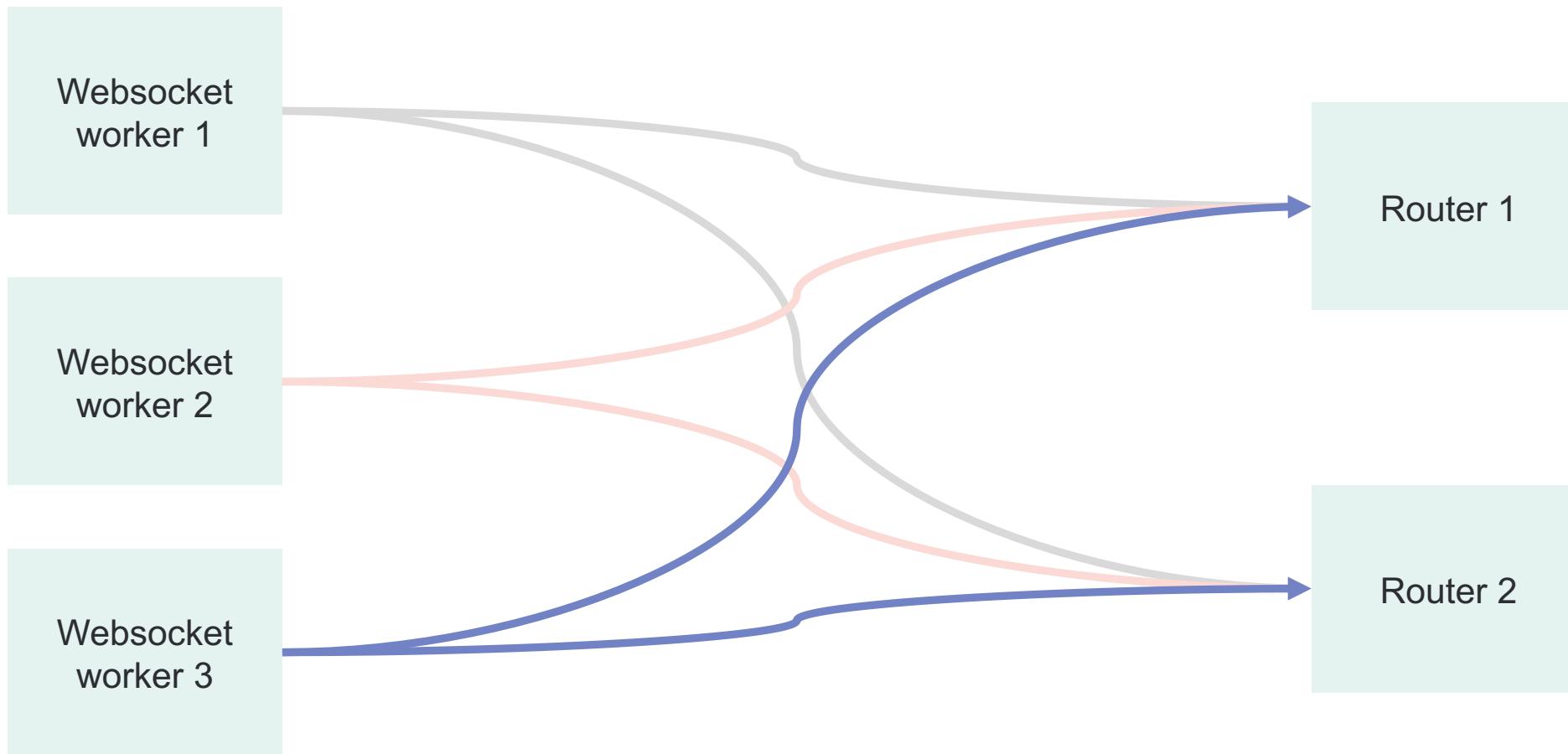


- Корутина получает нужное и отправляет пользователю



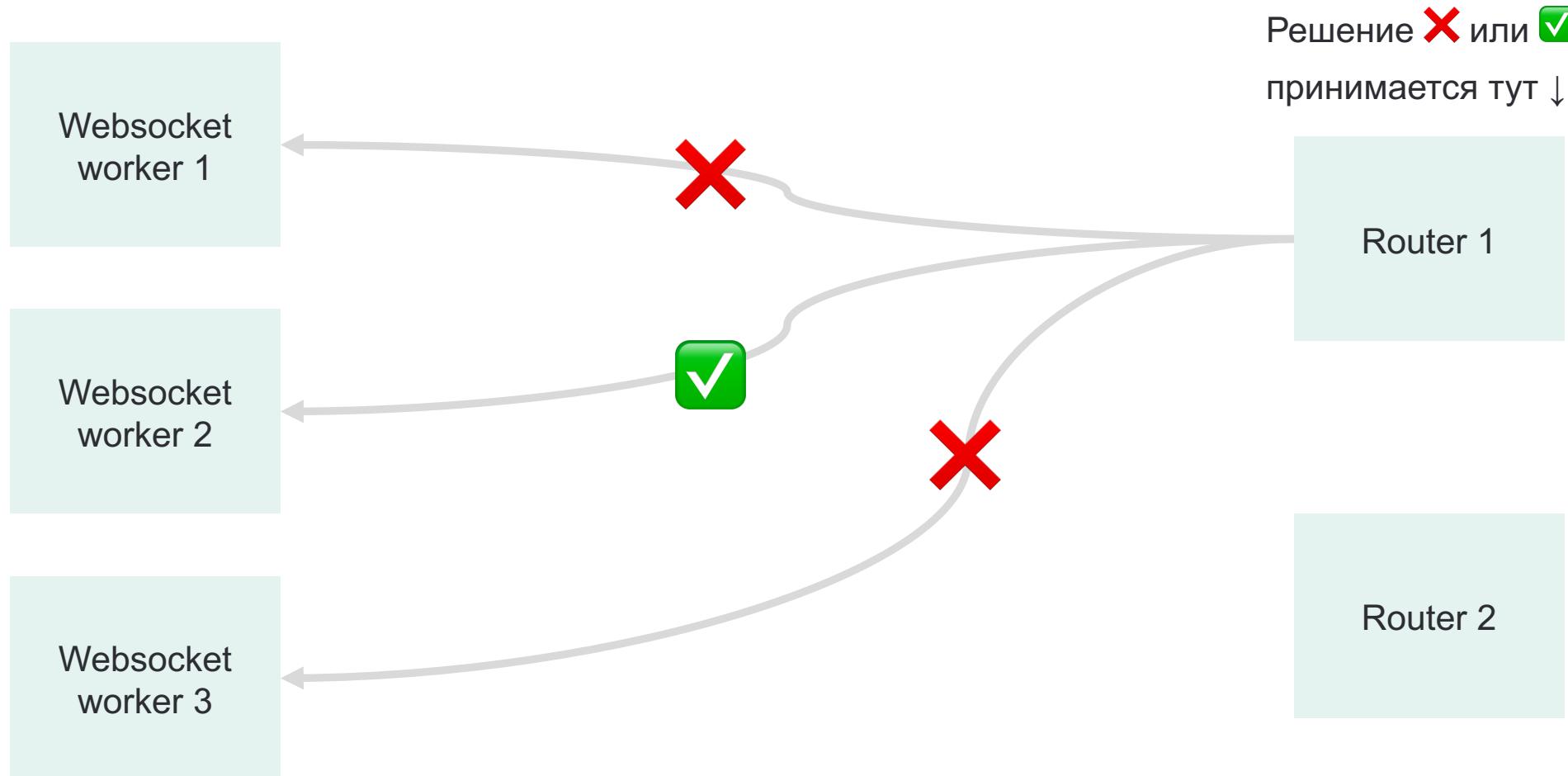
Как устроена подписка на роутеры

Сначала делаем это



Роутер pub'лишит в websocket «сервер» (корутину)

Потом вот это



X





Как устроен websocket сервер

```
async def wannabe_router(message_from_kafka: dict):
    zeromq_context = Context.instance()
    socket = zeromq_context.socket(PUB)
    socket.bind(f'tcp://{{ZERO_MQ_HOST}}:{{ZERO_MQ_PORT}}')
    await socket.send_multipart([b'user-id', json.dumps(message_from_kafka)])
```



Как устроен роутер

```
async def wannabe_websocket_server(websocket_client: WebSocket):
    zeromq_context = Context.instance()
    socket = zeromq_context.socket(SUB)
    socket.connect(f'tcp://{{ZERO_MQ_HOST}}:{{ZERO_MQ_PORT}}')
    socket.setsockopt(SUBSCRIBE, b'user-id')
    _, message = await socket.recv_multipart()
    await websocket_client.send_json(json.loads(message))
```

Планы на будущее

- добавить динамические ключи фильтрации в zeromq связку
- возможно сделать это с помощью «pub/sub» с фронтенда (иметь возможность sub'атся с фронтенда на websocket сервер)
- написать больше тестов 😊😊😊

Выводы по решению 2.0

- горизонтально масштабируется
- больше нет никакой инфра, только код
- бизнес доволен
- запах распределенного монолита прошёл

Выводы по решению 2.0

- ✓ горизонтально масштабируется
- ✓ больше нет никакой инфры, только код
- ✓ бизнес доволен
- ✓ запах распределенного монолита прошёл

- ✗ это сложно
- ✗ иногда нужна поясняющая бригада

Финальные выводы презентации

Возможно, вам подойдут готовые решения!

Финальные выводы презентации

Возможно, вам подойдут готовые решения!

Если нет, то:

— вы узнали как 2 способами решить проблему канала «вверх»

Финальные выводы презентации

Возможно, вам подойдут готовые решения!

Если нет, то:

- вы узнали как 2 способами решить проблему канала «вверх»
- у нас есть простой способ, который подойдет там, где нас не смущает шареный редис

Финальные выводы презентации

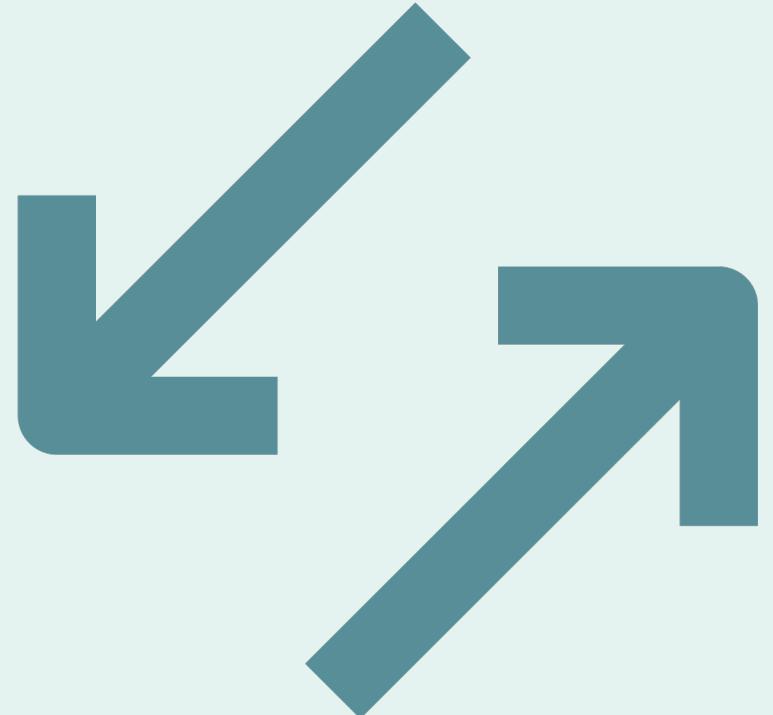
Возможно, вам подойдут готовые решения!

Если нет, то:

- вы узнали как 2 способами решить проблему канала «вверх»
- у нас есть простой способ, который подойдет там, где нас не смущает шареный редис
- а так же есть сложный способ подойдет там, где есть MSA, k8s и настоящее горизонтальное масштабирование



**Спасибо.
С радостью ответим на
вопросы!**



<https://linktr.ee/laukhin/>

<https://xfenix.ru/>