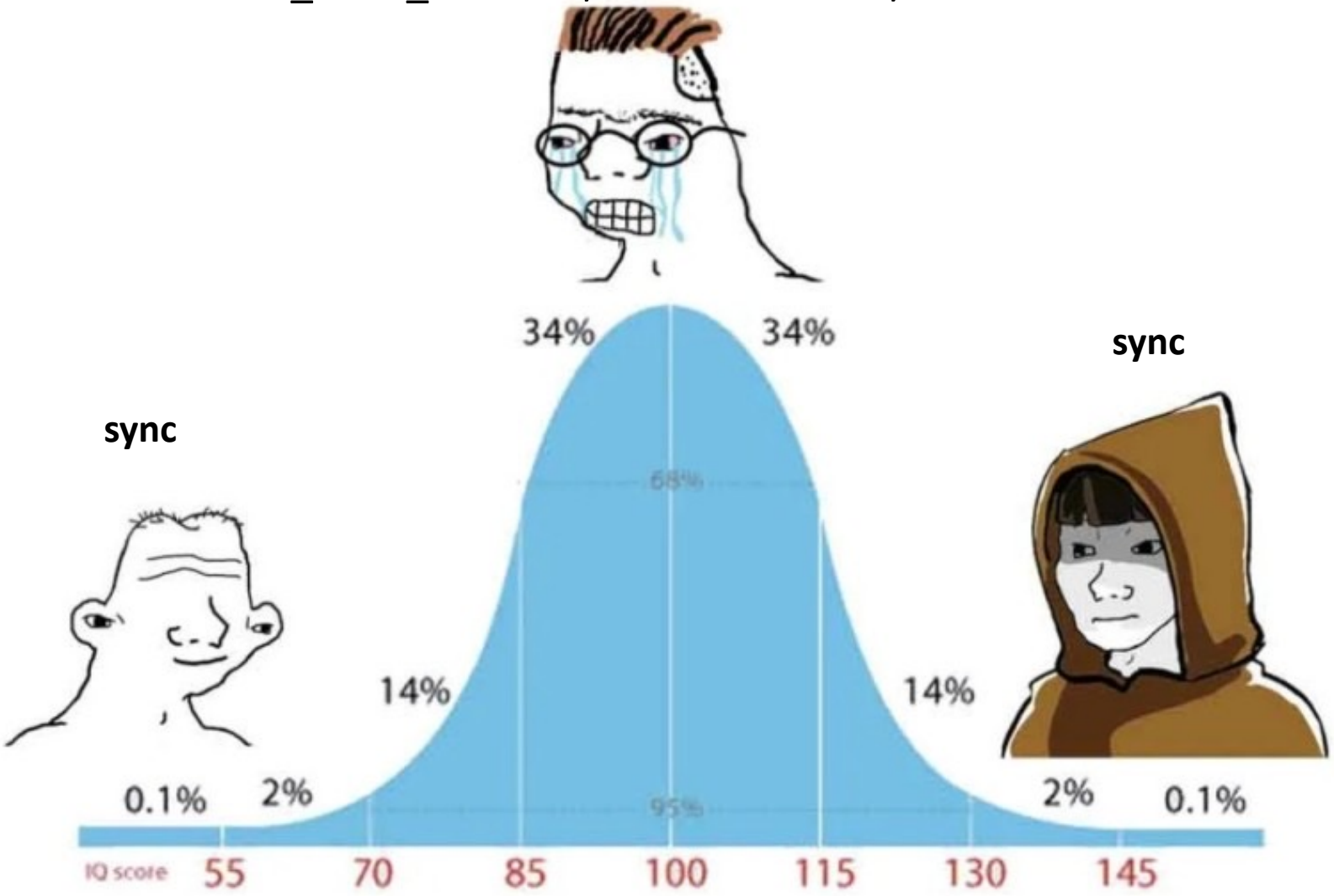
точка

банк для предпринимателей
и предприятий

Драматическое приручение *кролика*

Обо мне

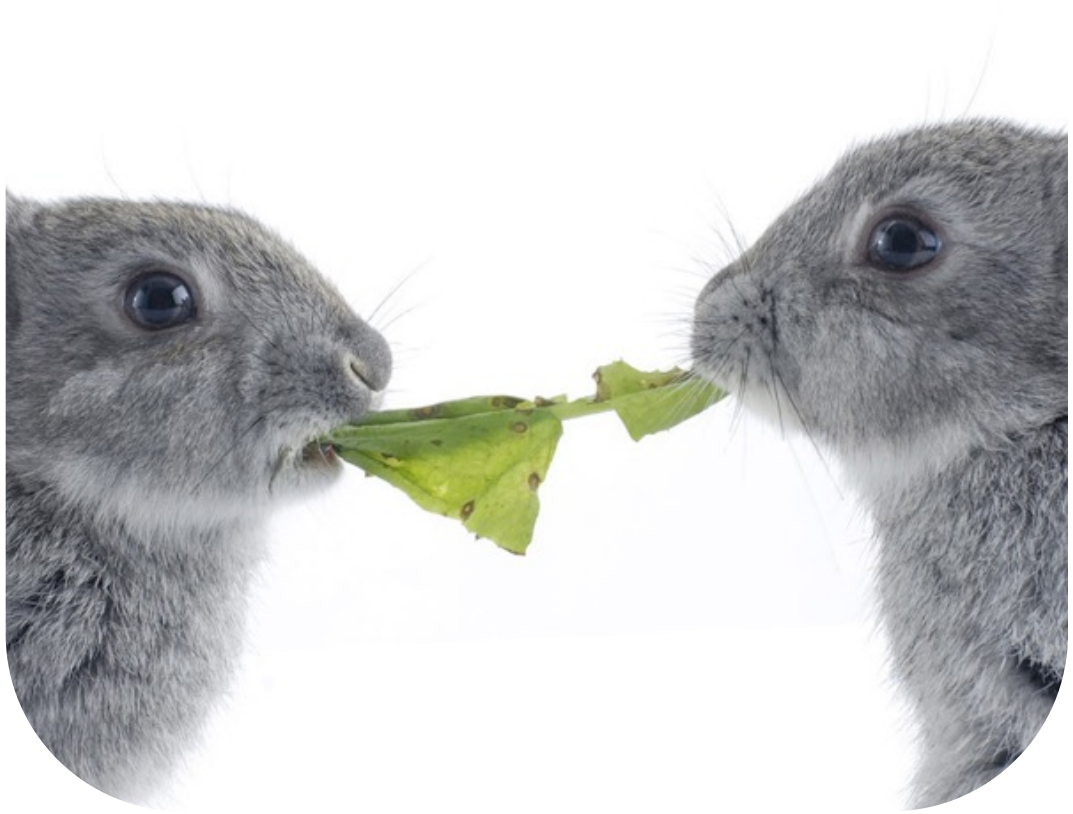
asyncio / create_task / “coroutine was never awaited” / add_done_callback / CancelledError / …



2013 2017 2018 2024

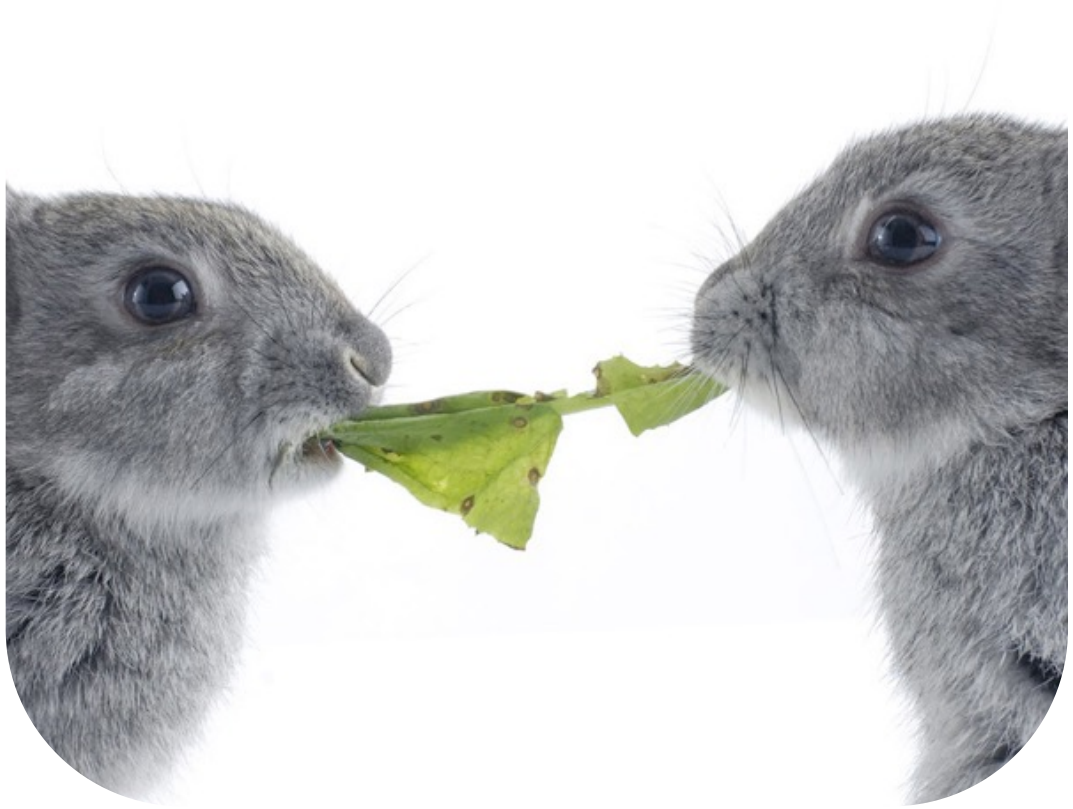
Celery

Сельдерей



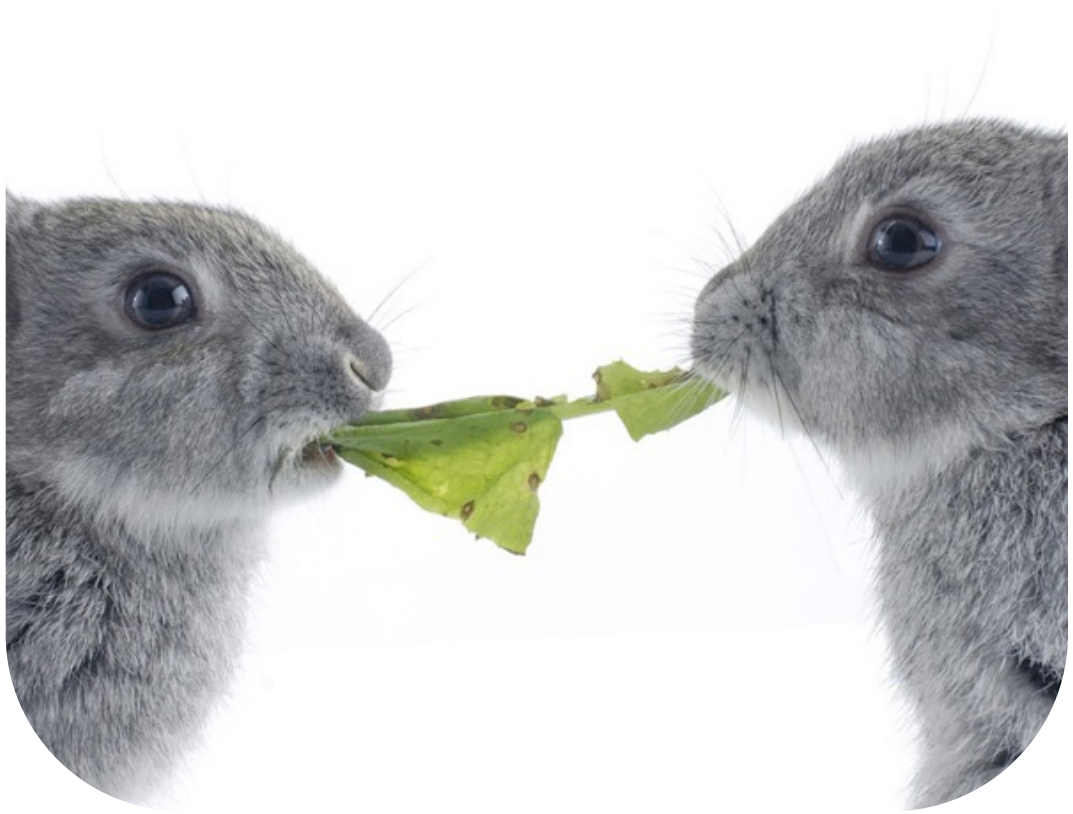
- Плохо с ретраями

Сельдерей



- Плохо с ретраями (не удобно)
- Нет DLQ (но сделать можно)

Сельдерей



- Плохо с ретраями (не удобно)
- Нет DLQ (но сделать можно)
- Надо уметь готовить

Сельдерей

Open | 2 tasks done

InterfaceError when using the celery_worker pytest fixture #4511

Audace opened this issue on Jan 30, 2018 · 26 comments

spumer commented on Sep 19, 2018 • edited

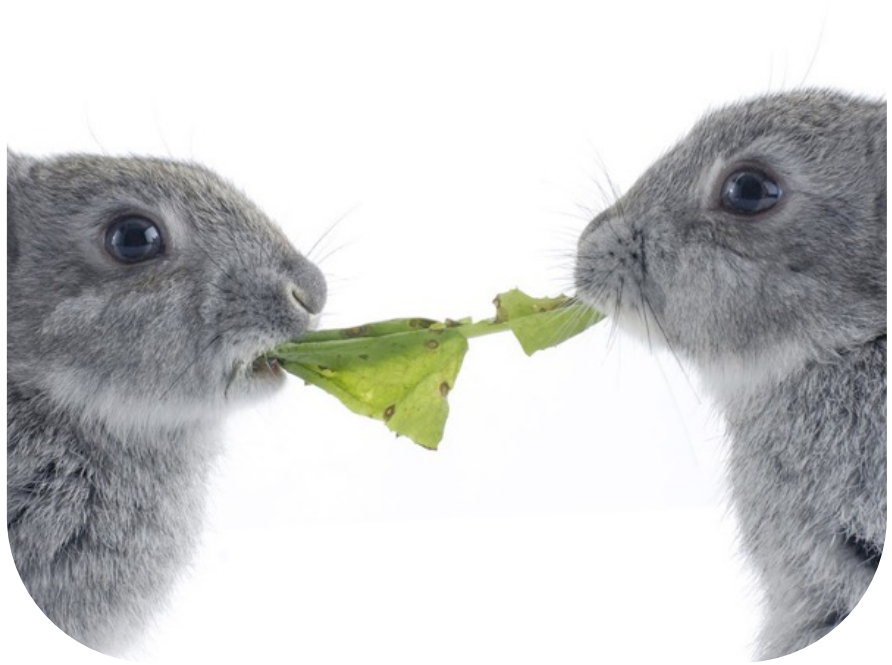
I found some workaround. This helps for me.

Some limitations of this method:

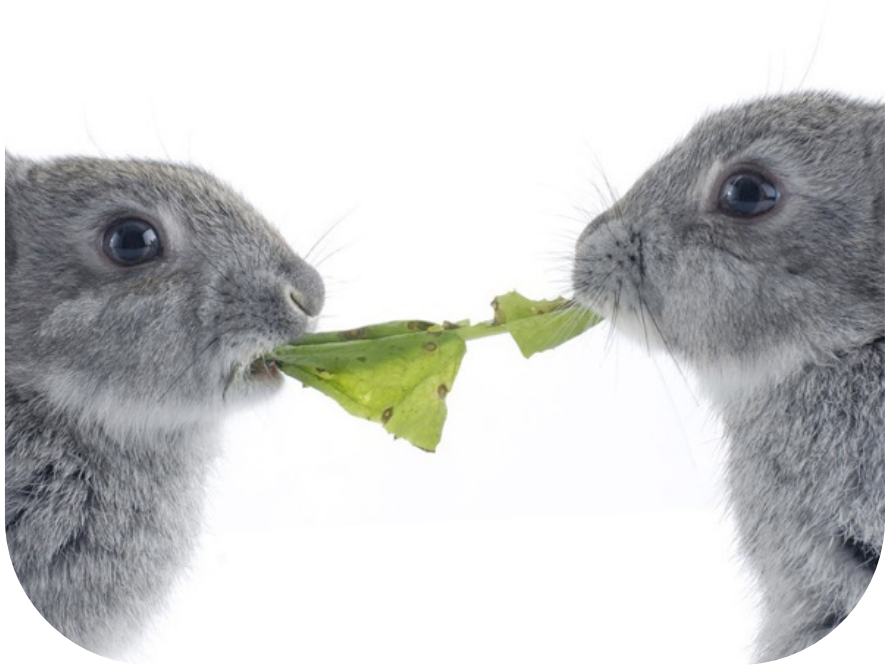
- Use session worker
- Cleanup broker after each test
- Dont use transactions for speedup tests

test_api.py

```
# !! THIS VERY IMPORTANT
# `transaction=True` disable transactions (atomics)
# and celery worker can see your test data
# Mark all tests in file
pytestmark = pytest.mark.django_db(transaction=True)

def test_ok(api_request, celery_worker_ex):
       # ... make api call trought `api_request` fixture-helper

      # we know one task should be called in that case
      celery_worker_ex.wait_task()

      # or we can check it explicit
      #task_result = celery_worker_ex.wait_task()
      #assert task_result.task_name = 'package.module.simple_task_name'

      # ... check result in DB or any other checks
```



Сельдерей



conftest.py

```python
from celery.contrib.testing.worker import TestWorkController
from celery.worker.request import Request


@pytest.fixture(autouse=True)
def celery_worker_ex(celery_session_worker: TestWorkController):
    w = celery_session_worker

    task_waiter = CeleryTaskWaiter(w.consumer)

    w.task_waiter = task_waiter

    w.wait_task = task_waiter.wait_task
    w.wait_tasks = task_waiter.wait_tasks

    w._setup_hook = task_waiter._setup_hook
    w._remove_hook = task_waiter._remove_hook

    yield w

    task_waiter.cleanup()


def _hook_bound_func(obj, name, hook):
    """Create pre-hook of bound method"""
    orig = getattr(obj, name)

    def wrapped(self, *args, **kw):
        hook(self, *args, **kw)
        return orig(*args, **kw)

    setattr(obj, name, types.MethodType(wrapped, obj))
    setattr(obj, f'{name}__original', orig)


def _unhook_bound_func(obj, name):
    orig = getattr(obj, f'{name}__original')
```

Dramatiq

Dramatiq: то же самое, только лучше

Dramatiq: то же самое, только лучше

- ЕСТЬ middleware (Delay/Retry из коробки)

Dramatiq: то же самое, только лучше

- ЕСТЬ middleware (Delay/Retry из коробки)
- ЕСТЬ DLQ (dead-letter)

Dramatiq: то же самое, только лучше

- ЕСТЬ middleware (Delay/Retry из коробки)
- ЕСТЬ DLQ (dead-letter)
- Легок в тестировании и эксплуатации

Dramatiq: то же самое, только лучше

- ЕСТЬ middleware (Delay/Retry из коробки)
- ЕСТЬ DLQ (dead-letter)
- Легок в тестировании и эксплуатации
- НЕ овощ и НЕ зелень

Dramatiq: то же самое, только лучше

Dramatiq: то же самое, только лучше

- Task Execution Time Limit

Dramatiq: то же самое, только лучше

- Task Execution Time Limit
- Asyncio (async def)

Dramatiq: то же самое, только лучше

- Task Execution Time Limit
- Asyncio (async def)
- И много ещё другого:

https://dramatiq.io/motivation.html

FastStream

Sep 18, 2023

sternakt

0.1.0

6bd05df

Compare

v0.1.0

FastStream is a new package based on the ideas and experiences gained from FastKafka and Propan. By joining our forces, we picked up the best from both packages and created the unified way to write services capable of processing streamed data regardless of the underlying protocol. We'll continue to maintain both packages, but new development will be in this project. If you are starting a new service, this package is the recommended way to do it.

Features


A FEW
YEARS LATER

Барон фон Таскинг

Барон и его очереди

Барон и его очереди

- 37 фоновых задач

Барон и его очереди

- 37 фоновых задач
- 300k / сут. (~3.5rps)

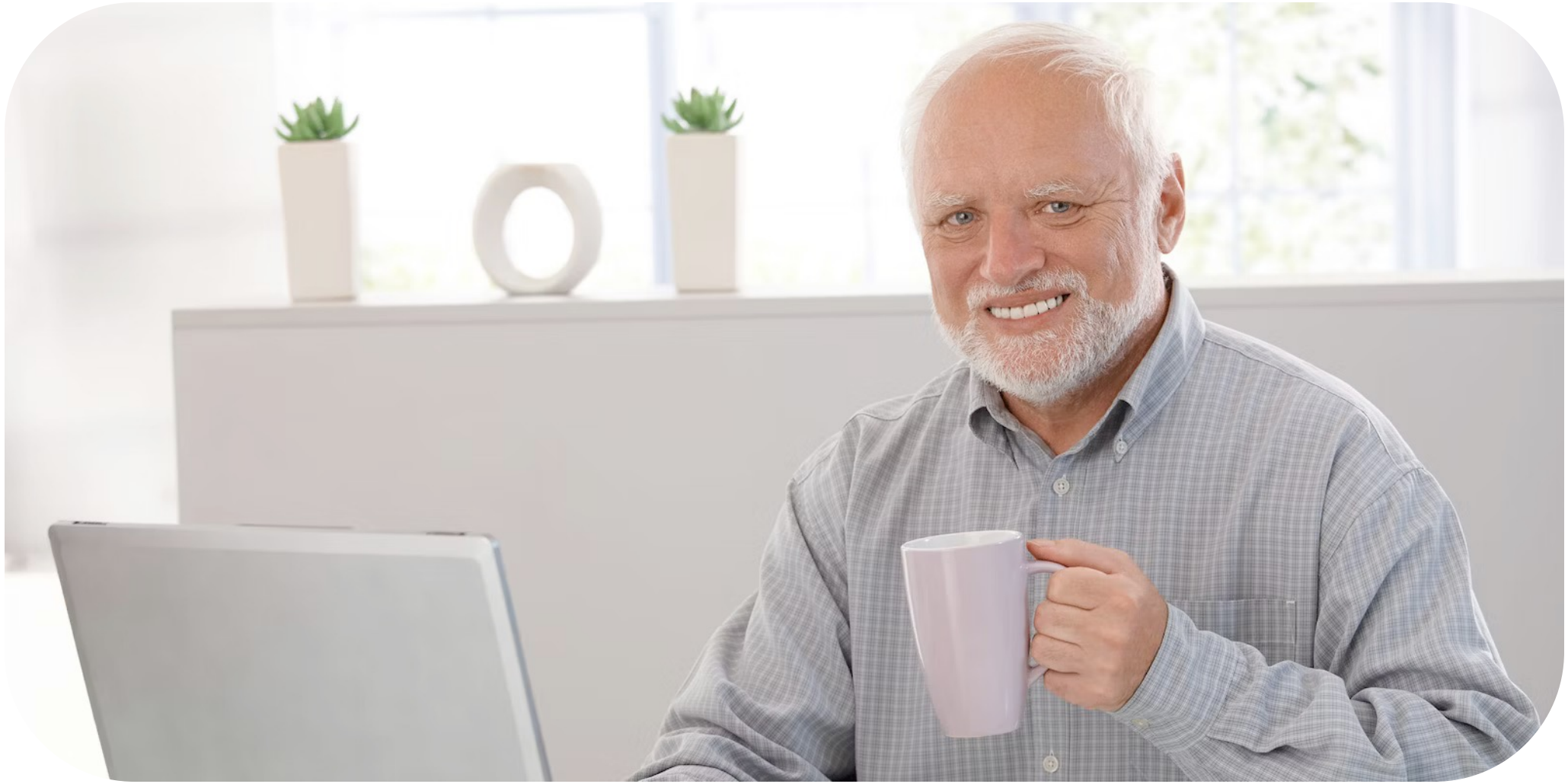
Барон и его очереди

- 37 фоновых задач
- 300k / сут. (~3.5rps)
- Приоритезаиця:
  2 очереди

Барон и его очереди

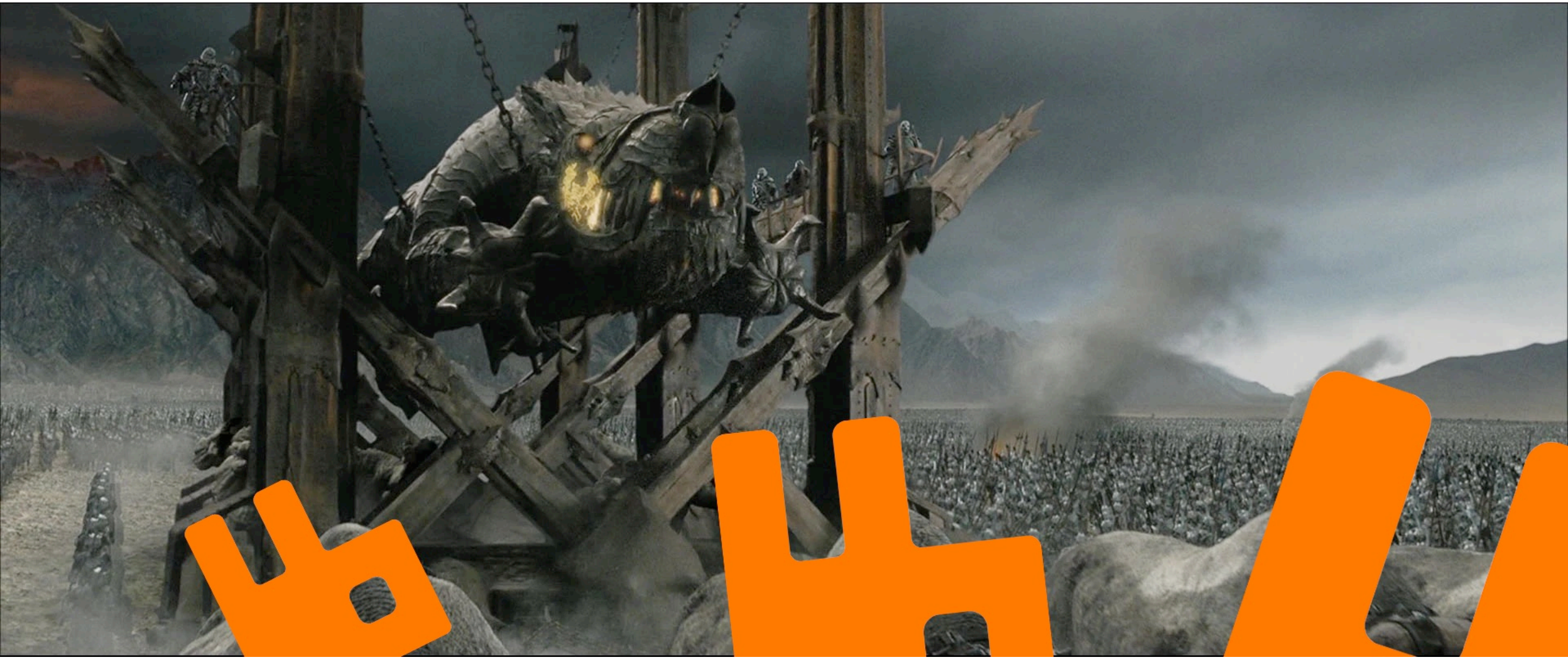
- 37 фоновых задач
- 300k / сут. (~3.5rps)
- Приоритезаиця:
  2 очереди
- 6 инстансов dramatiq

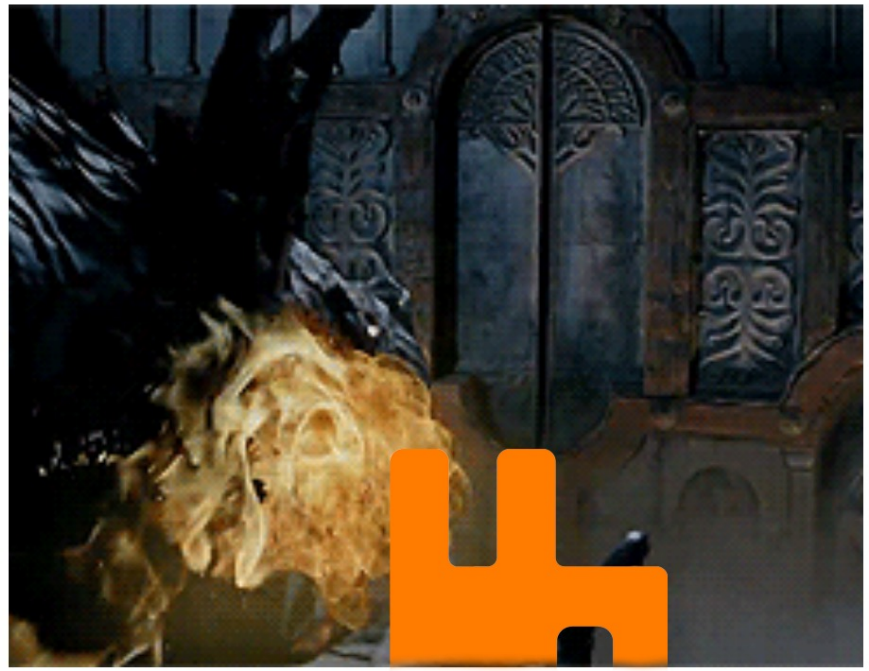
Барон и его очереди



Барон и его очереди

Барон на очереди




4


Access to vhost refused: ...

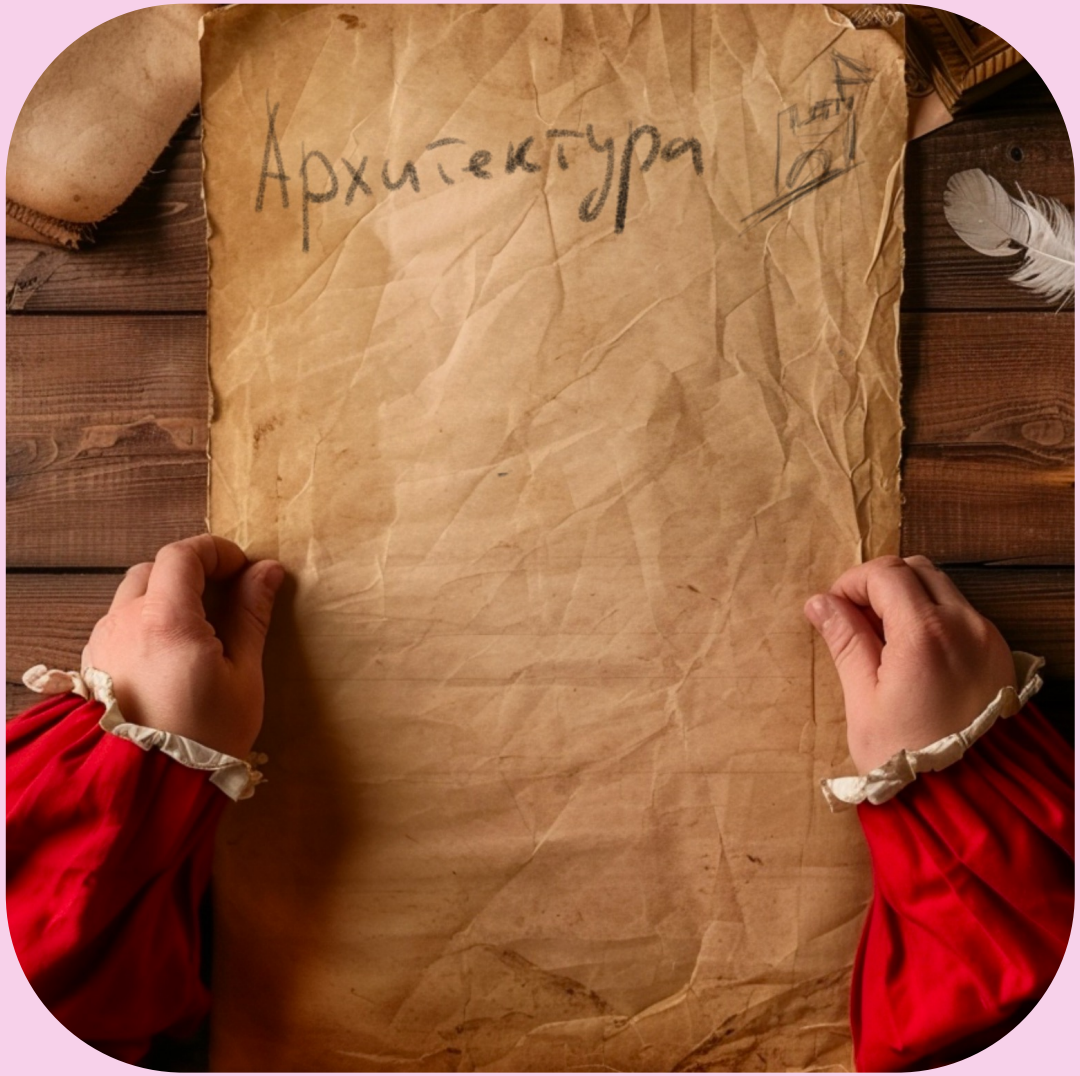

Connection limit Reached


EXCEPTION
EXCEPTION
EXCEPTION
limetraed

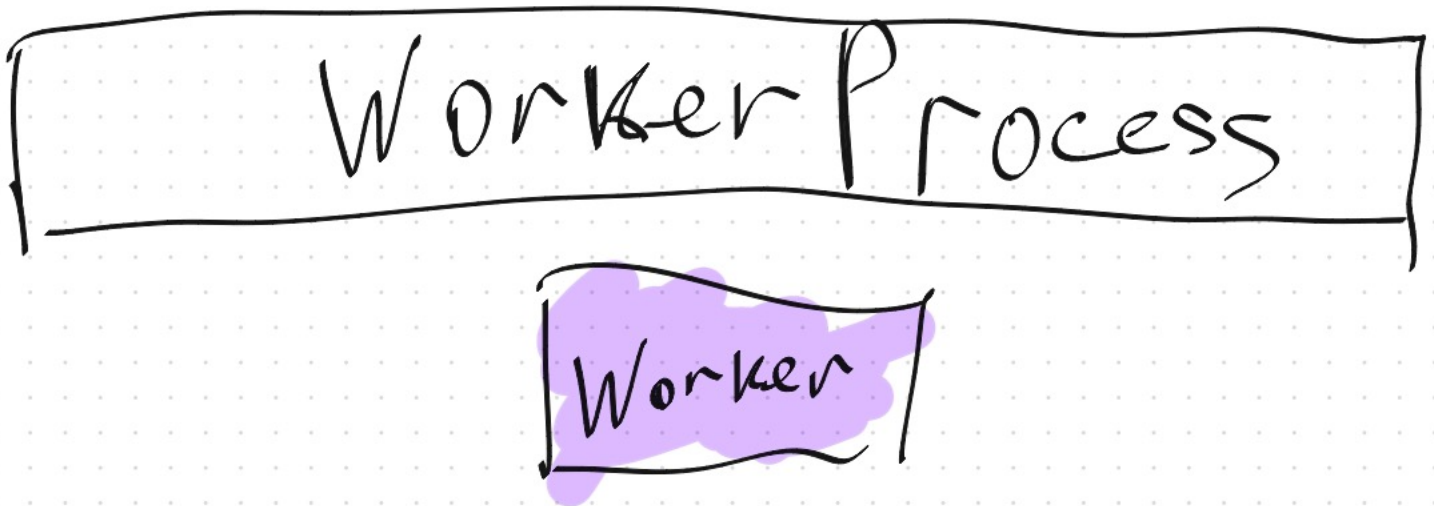
Internals

Архитектура

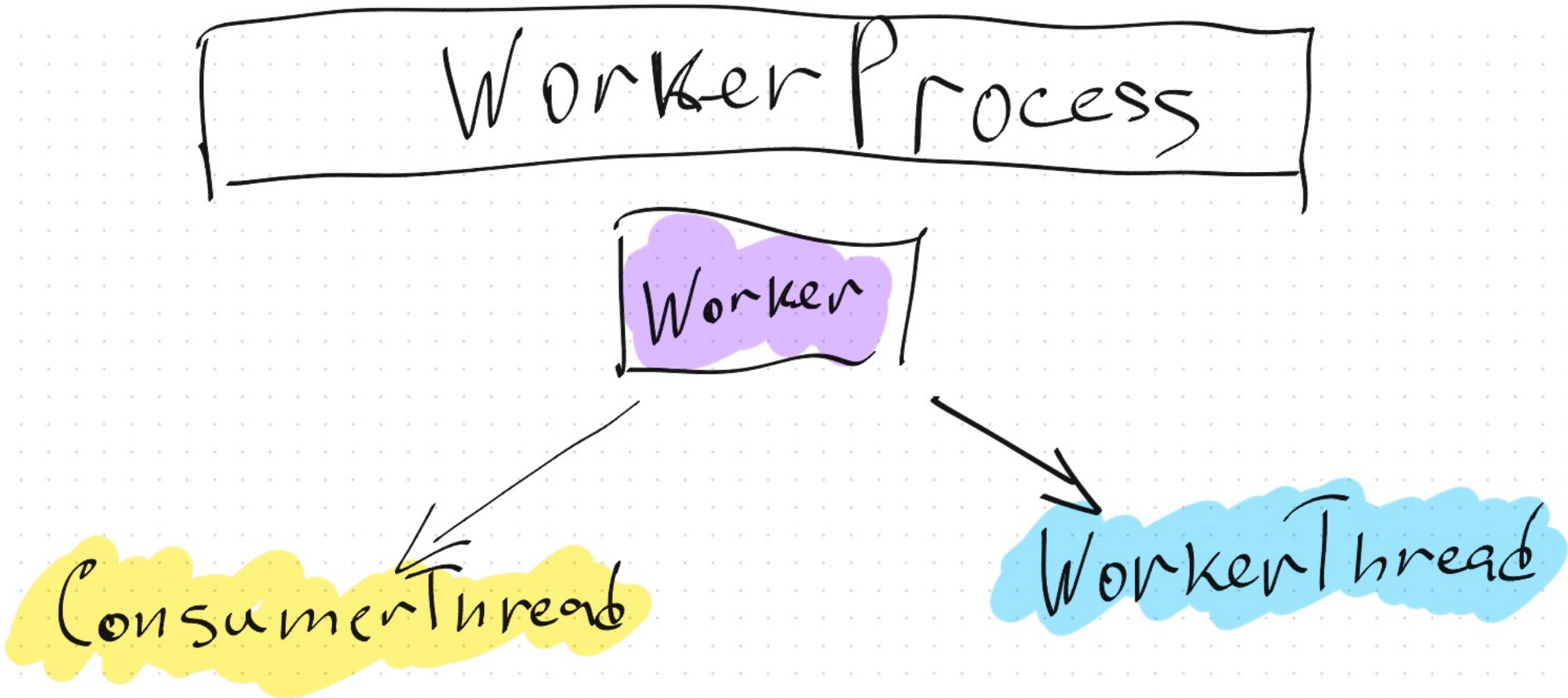
Dramatiq



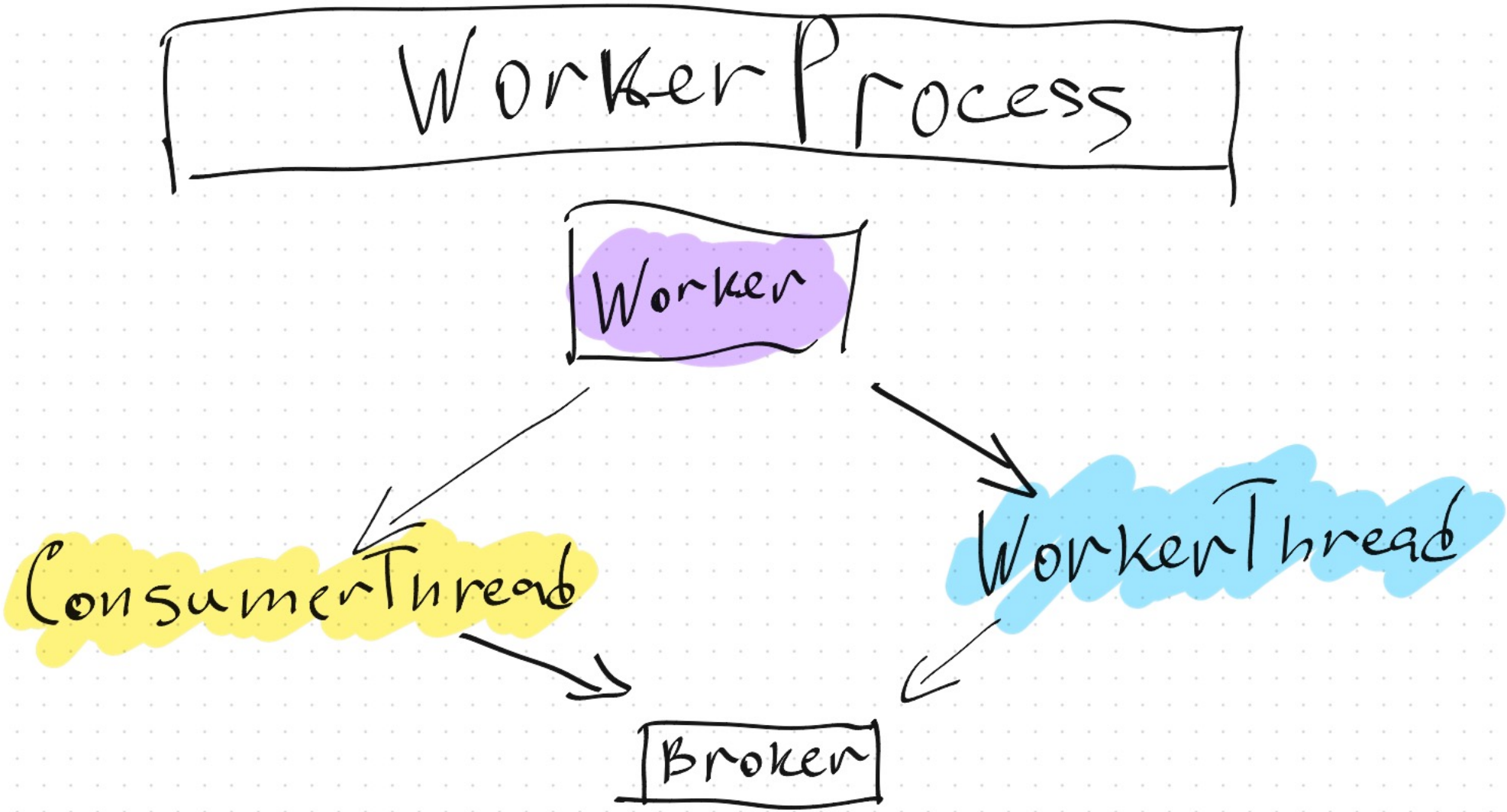
Dramatiq internals



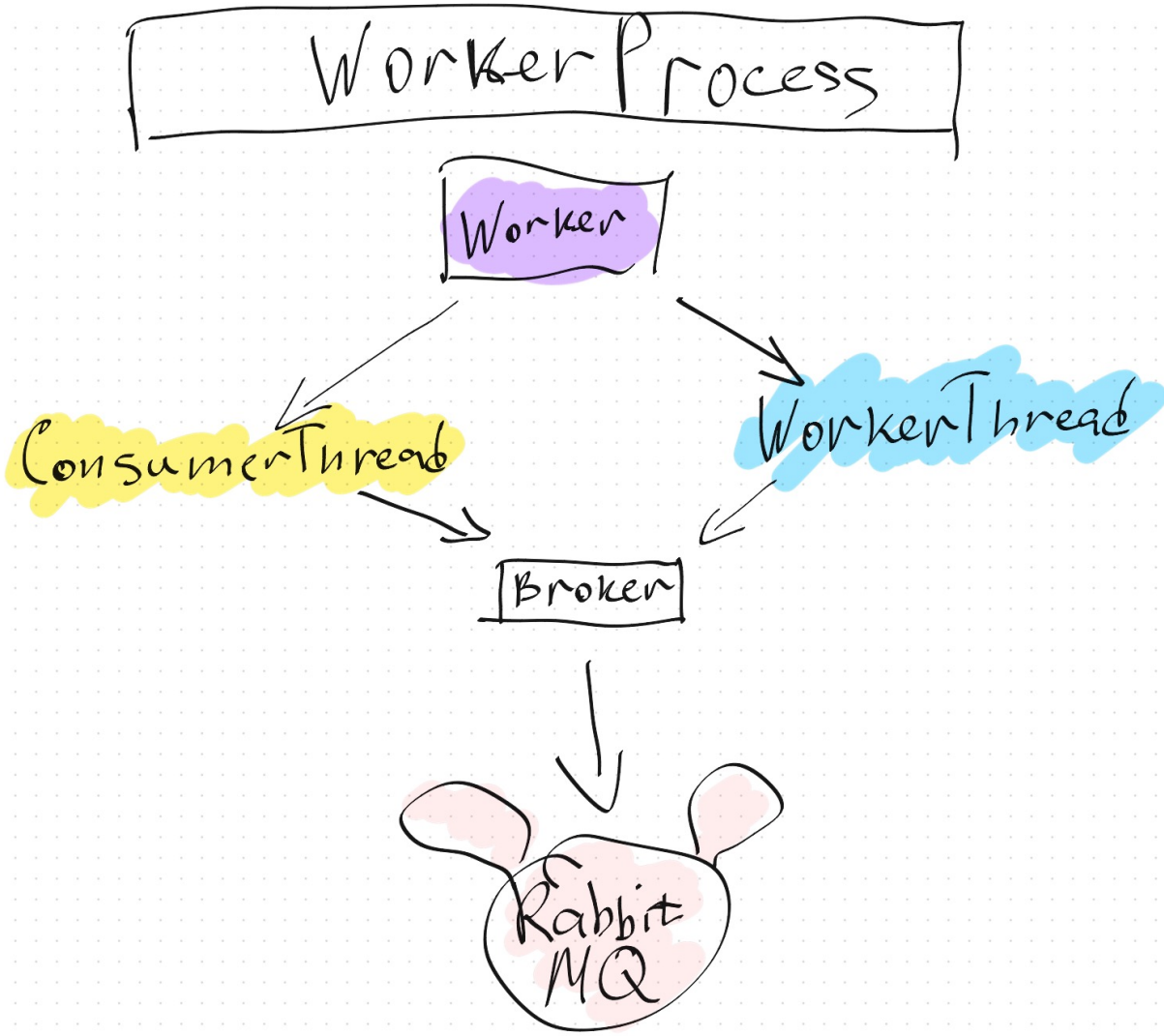
Dramatiq internals



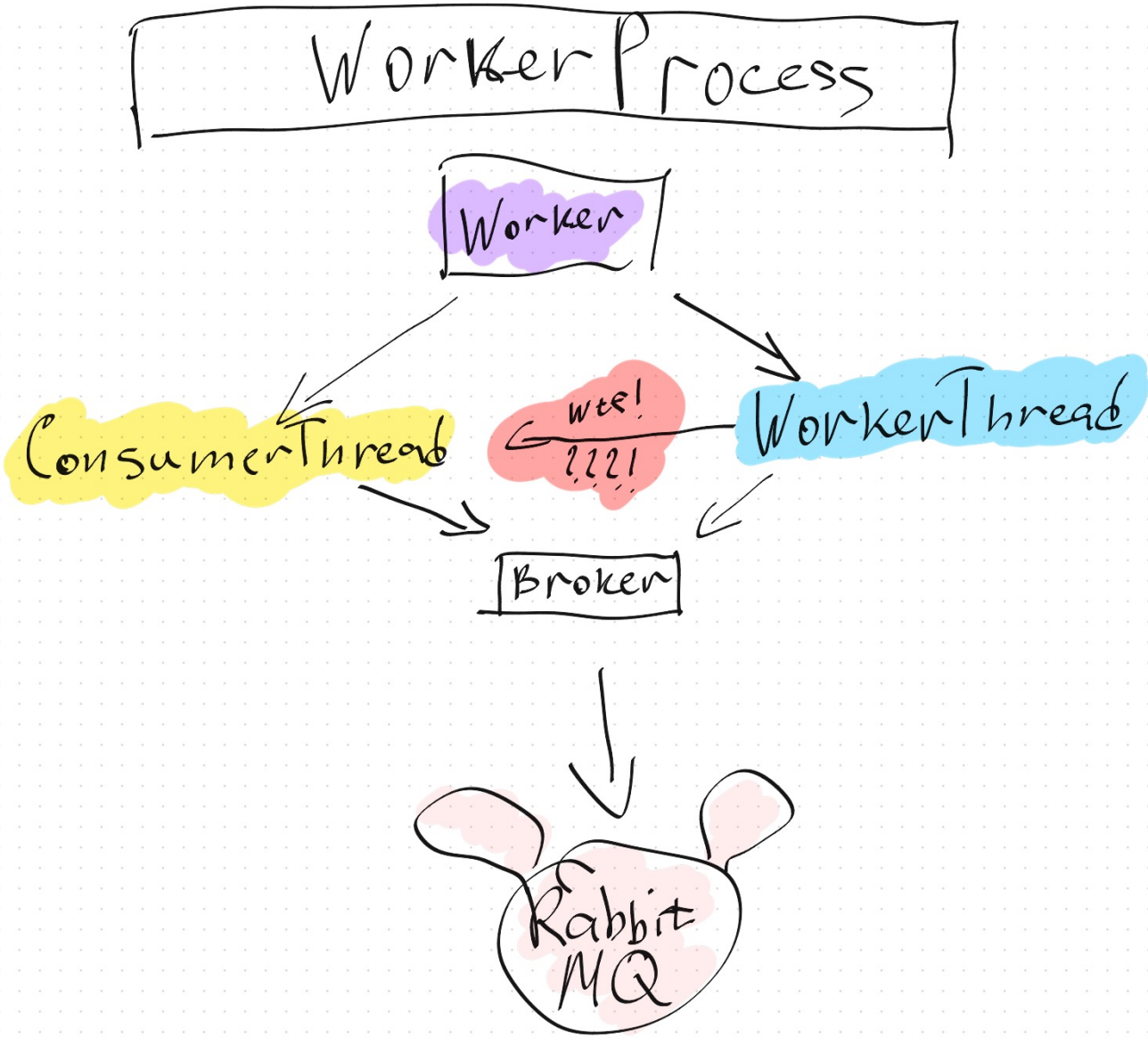
Dramatiq internals



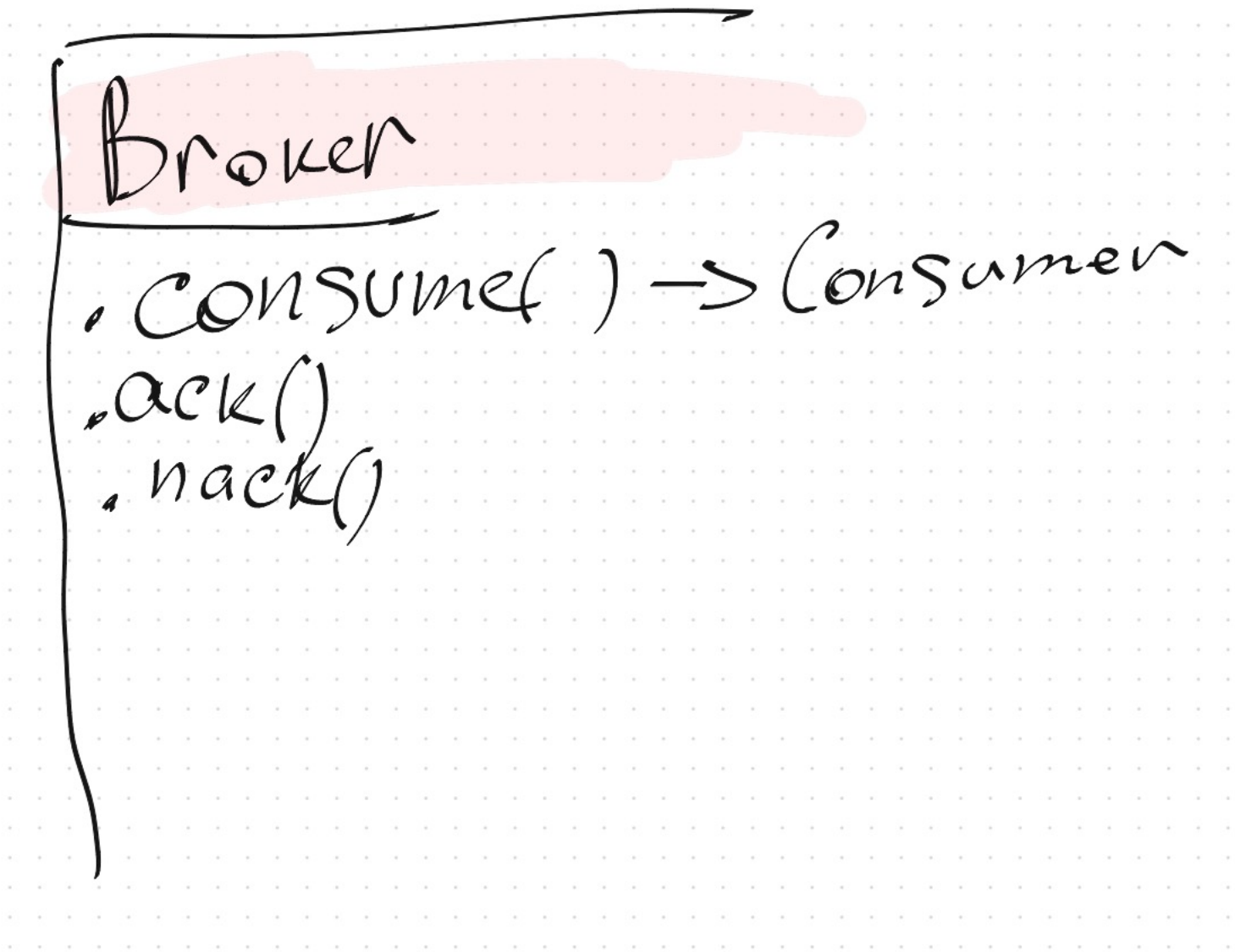
Dramatiq internals



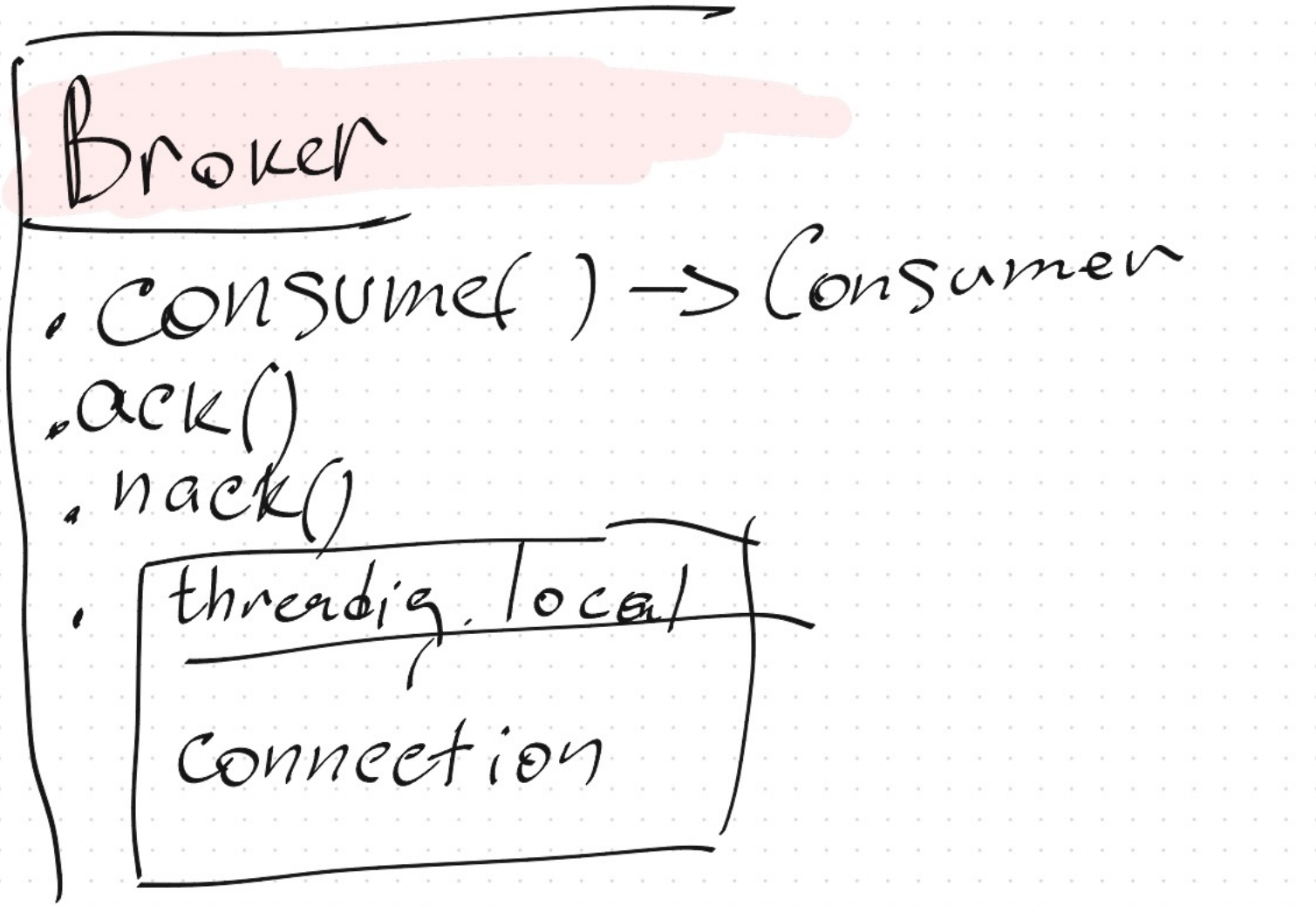
Dramatiq internals



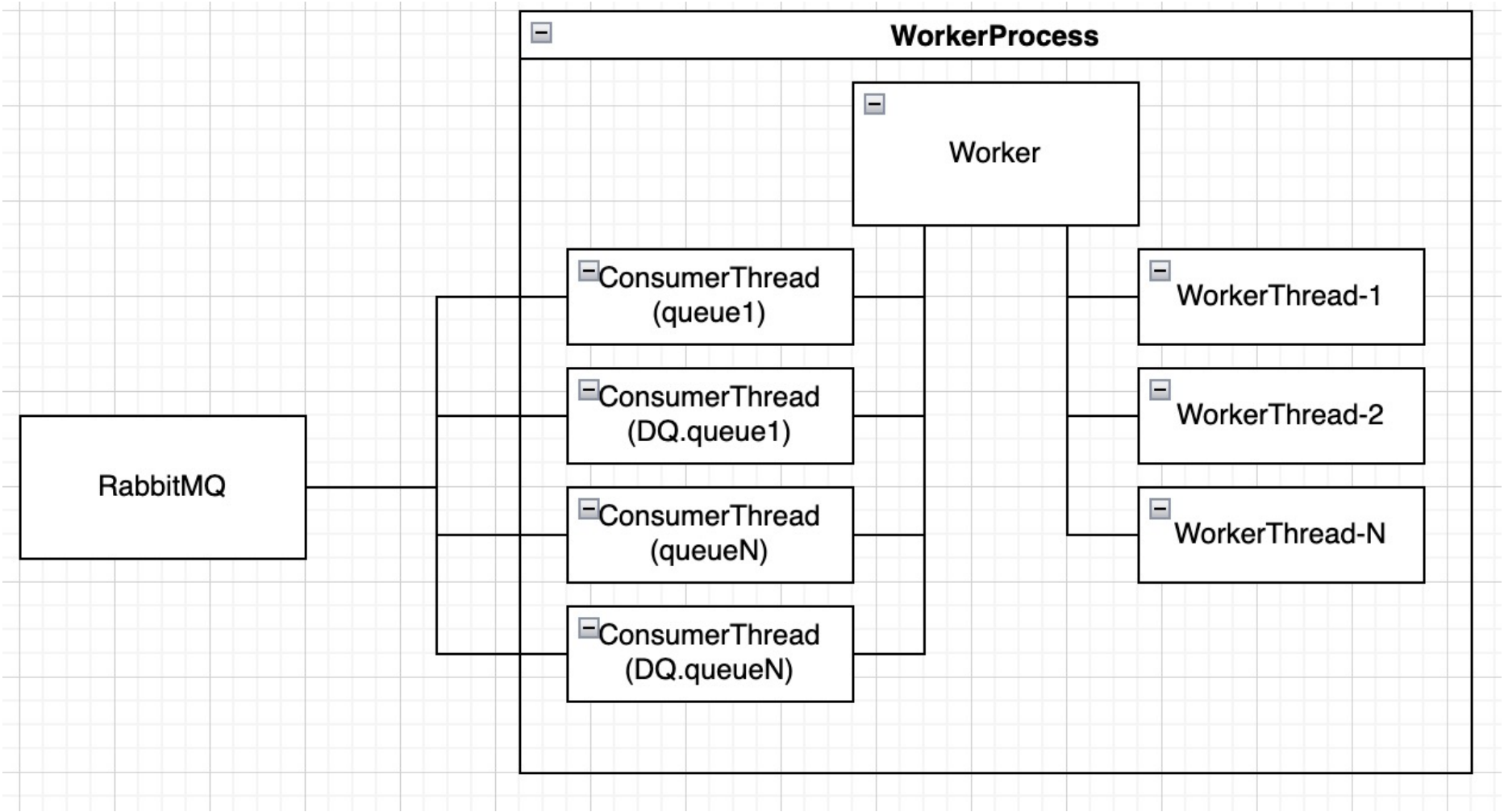
Dramatiq internals



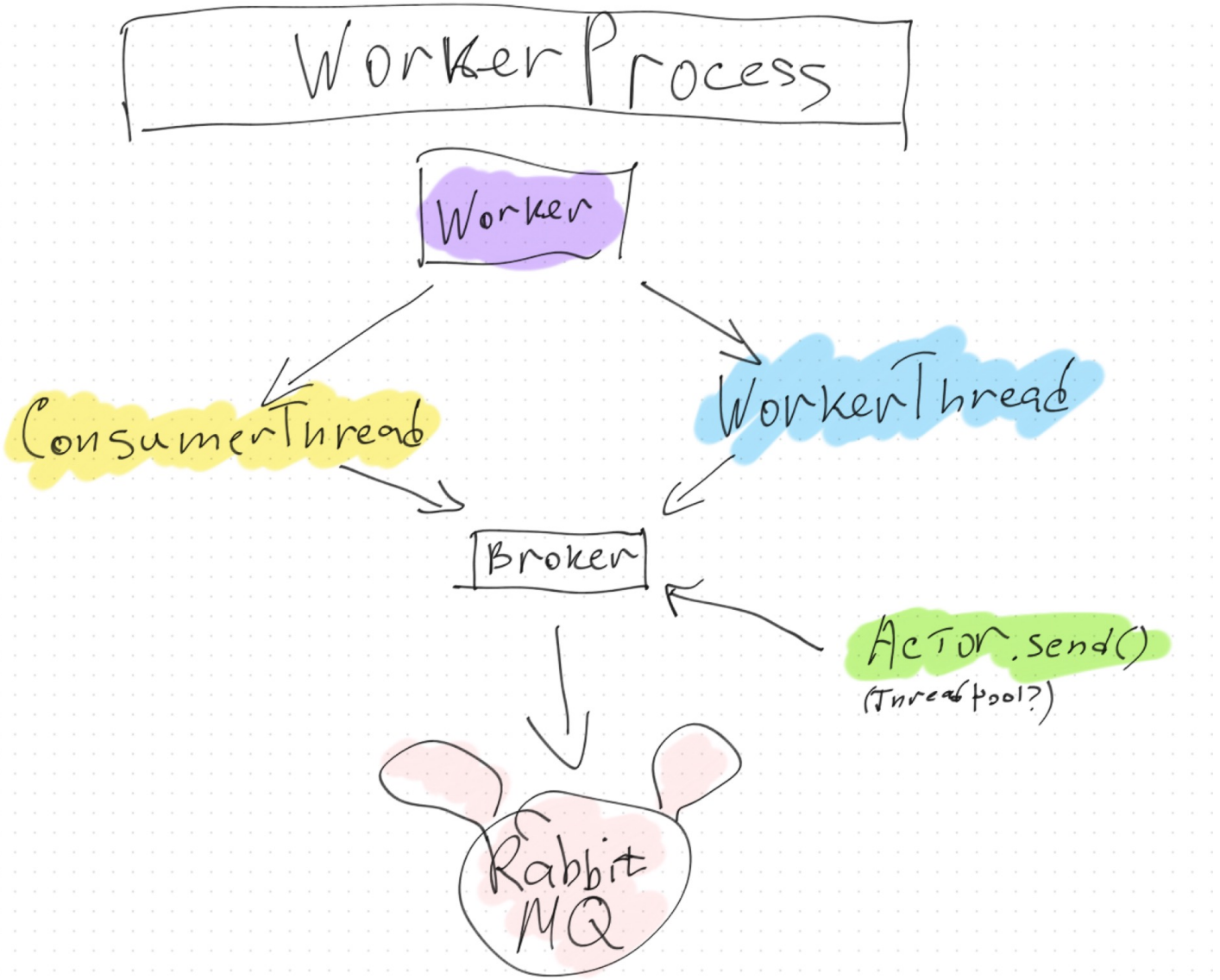
Dramatiq internals



Dramatiq internals



Dramatiq internals



Dramatiq internals

Bogdanp commented on Jul 4, 2019 Owner

You have two options:

- either you can increase your heartbeat timeout (`RabbitmqBroker(heartbeat=600)`) to >1m (it looks like yours is currently 5s from the rmq logs, which is unusual) or
- close the connection after every send via `get_broker().connection.close()` .


RabbitMQ
Developer Center

RabbitMQ Best practice

Single TCP

RabbitMQ: Single TCP

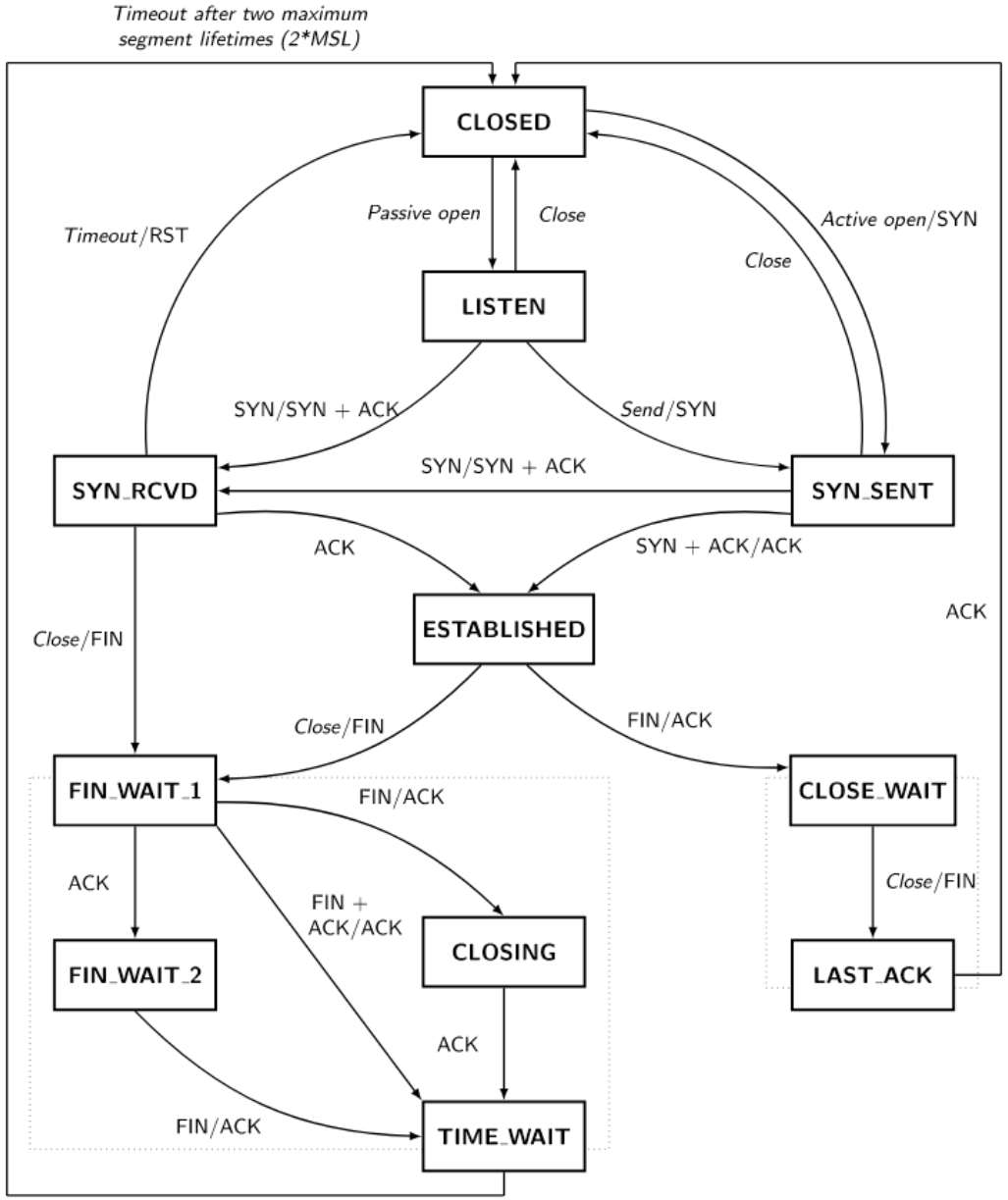
RabbitMQ: Single TCP

- Меньше соединений – лучше

RabbitMQ: Single TCP

- Меньше соединений – лучше
- TCP создан чтобы жить долго

RabbitMQ: Single TCP



https://vincent.bernat.ch/en/blog/2014-tcp-time-wait-state-linux

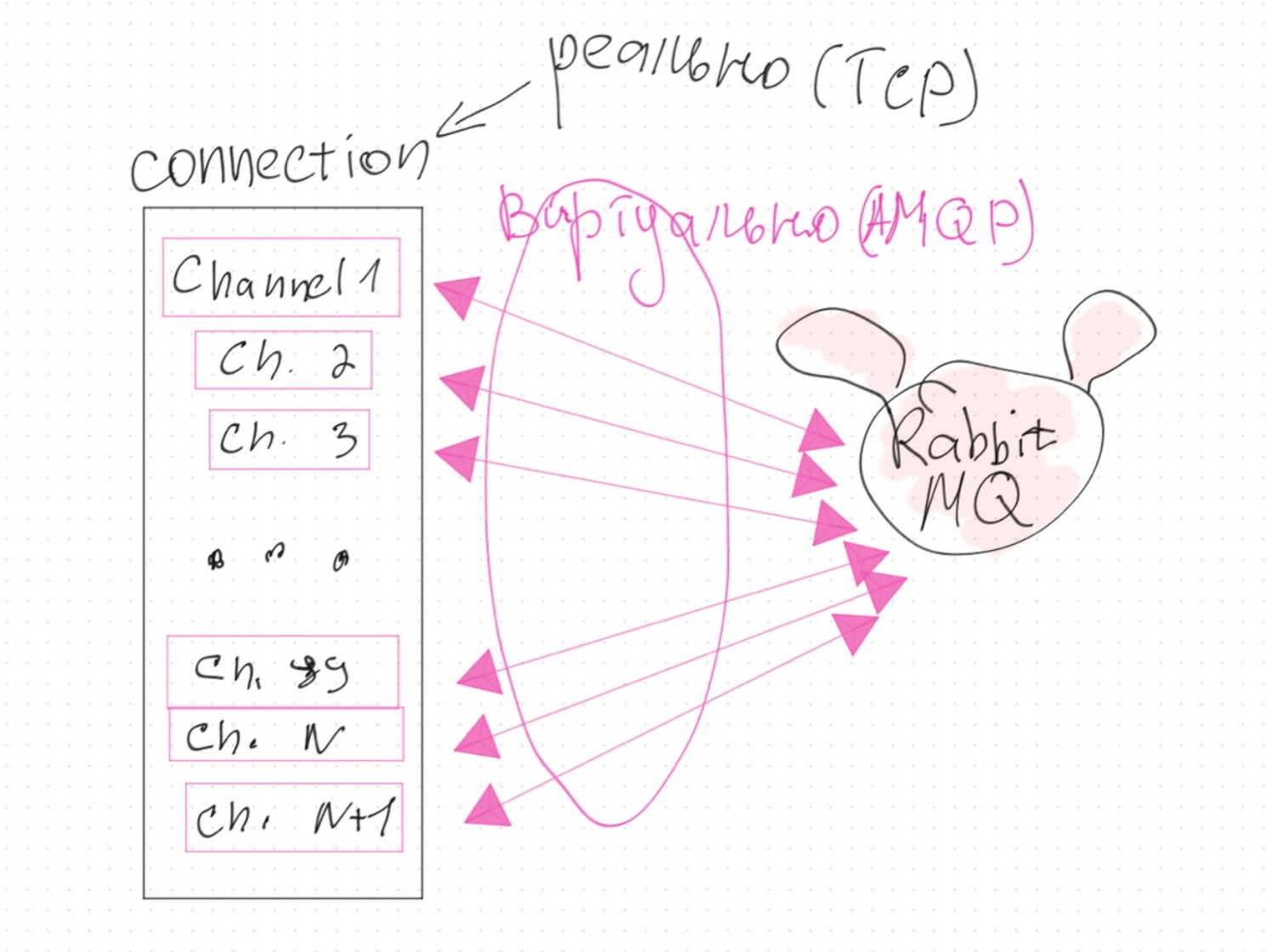
RabbitMQ: Single TCP

- Меньше соединений – лучше
- TCP создан чтобы жить долго
- Используйте каналы

RabbitMQ

AMQP over TCP

RabbitMQ: AMQP over TCP



RabbitMQ: AMQP over TCP

- 1-connection-N-channel

RabbitMQ: AMQP over TCP

- 1-connection-N-channel

RabbitMQ: AMQP over TCP

- 1-connection-N-channel
- Каналы дешевые

RabbitMQ: AMQP over TCP

- 1-connection-N-channel
- Каналы дешевые
- Используются всегда


NEWS

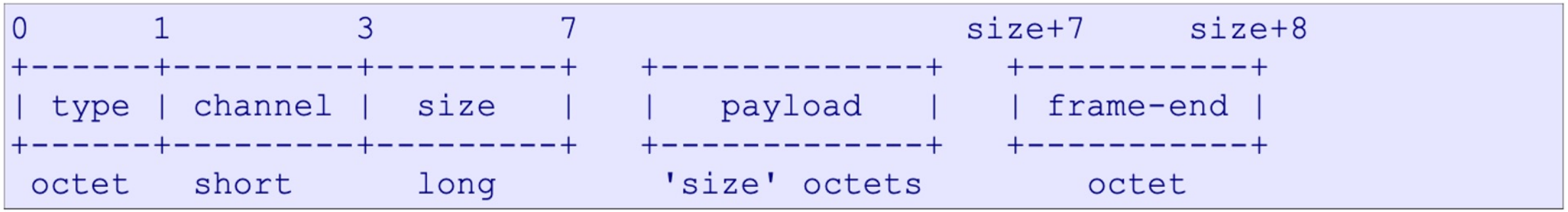
RabbitMQ: News

- AMQP Framed

Rabbit News: AMQP Framed

4.2.3 General Frame Format

All frames start with a 7-octet header composed of a type field (octet), a channel field (short integer) and a size field (long integer):



https://www.rabbitmq.com/resources/specs/amqp0-9-1.pdf

RabbitMQ: News

- AMQP Framed
- AMQP это RPC

RabbitMQ: News

- AMQP Framed
- AMQP это RPC
  - “Открой канал”
  - “Закрой канал”

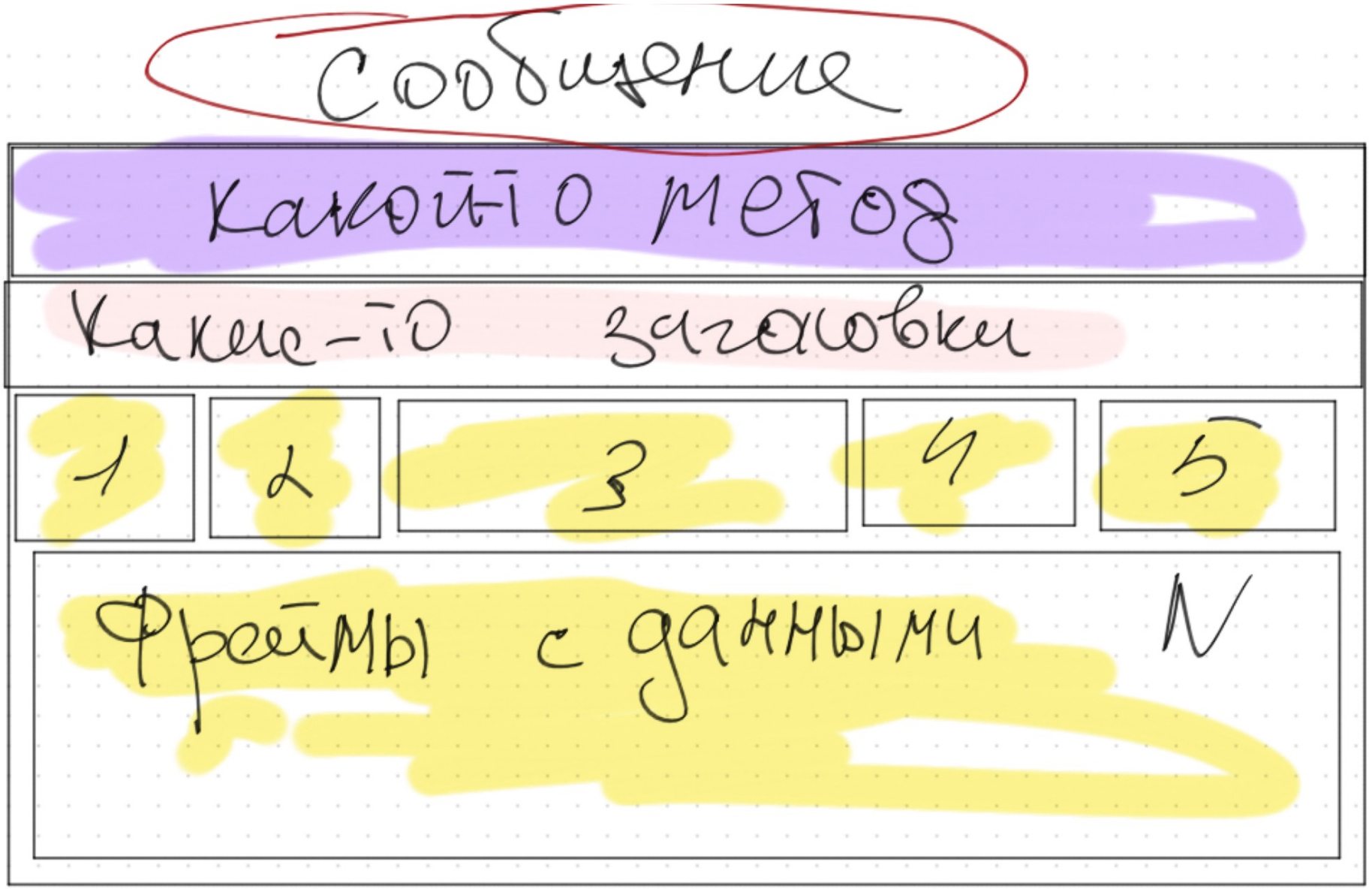
RabbitMQ: News

- AMQP Framed
- AMQP это RPC
  - “Открой канал”
  - “Закрой канал”
  - “Создай очередь”
  - “Удали очередь”

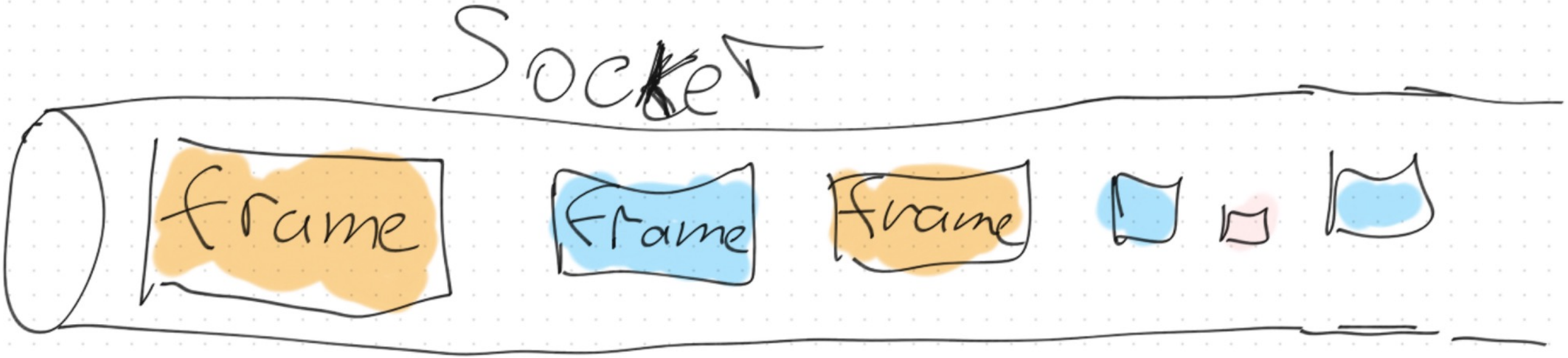
RabbitMQ: News

- AMQP Framed
- AMQP это RPC
  - “Открой канал”
  - “Закрой канал”
  - “Создай очередь”
  - “Удали очередь”
  - “Добавь в очередь” — publish

RabbitMQ: AMQP RPC

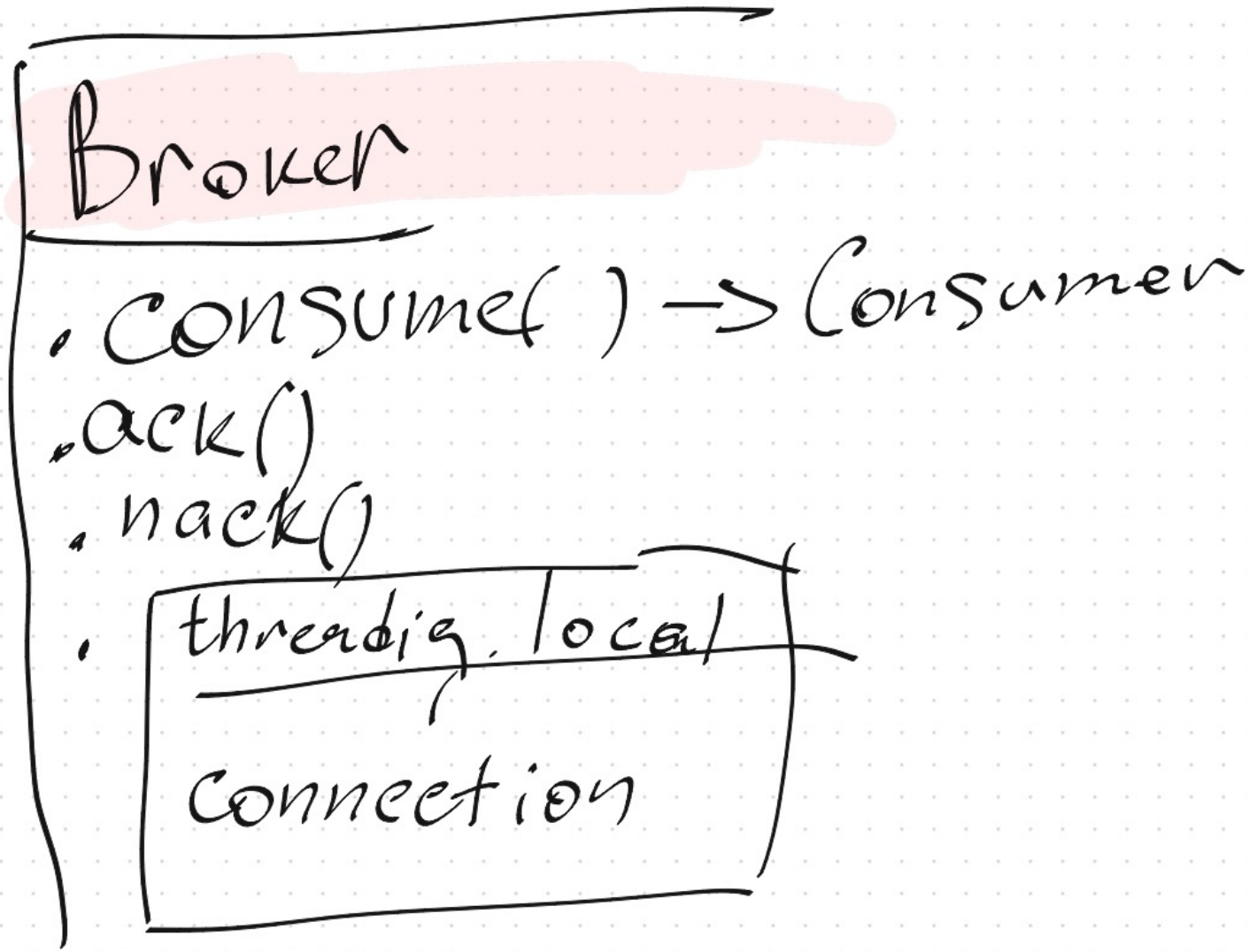


RabbitMQ: AMQP RPC

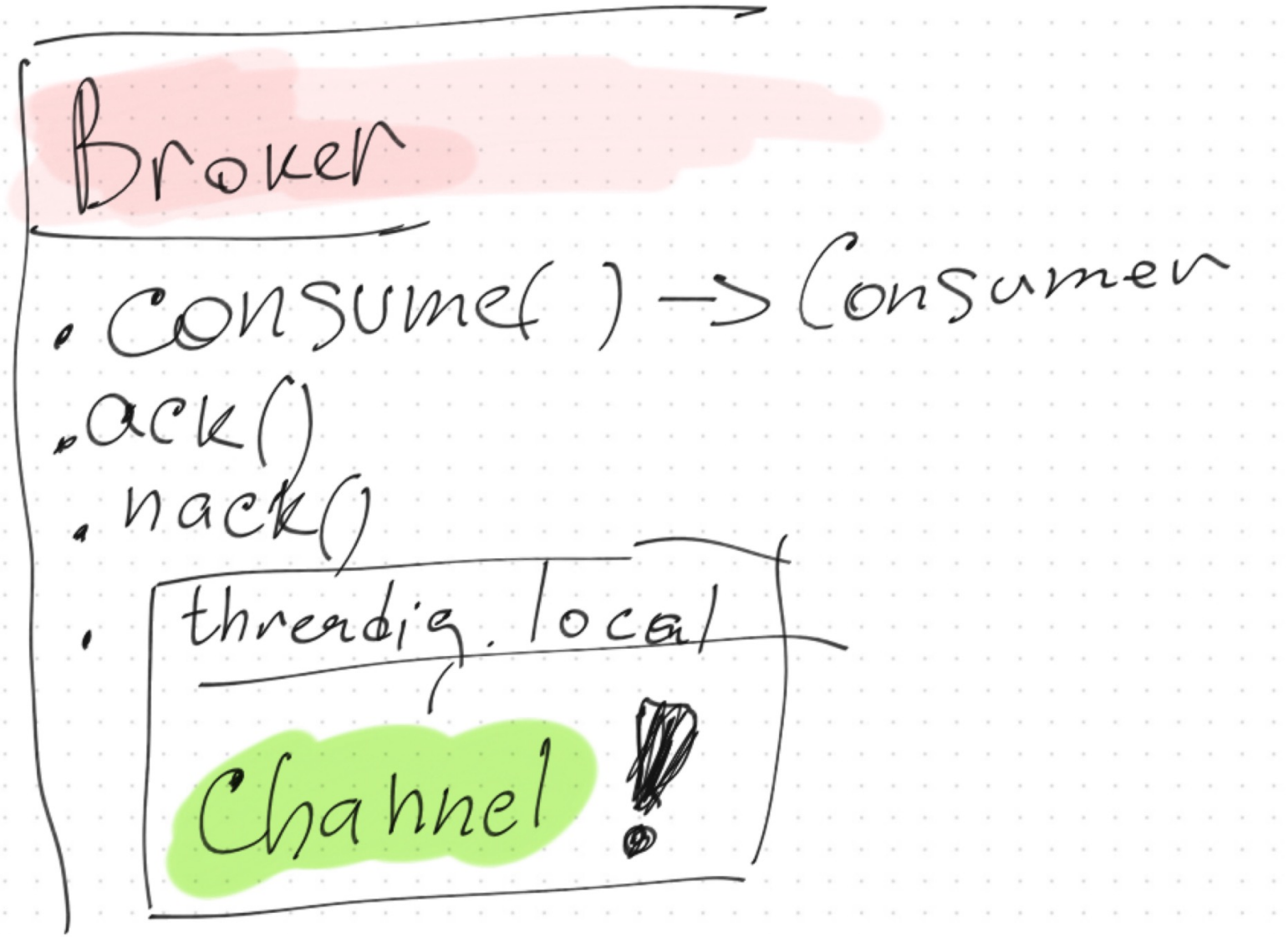


Last *Step*

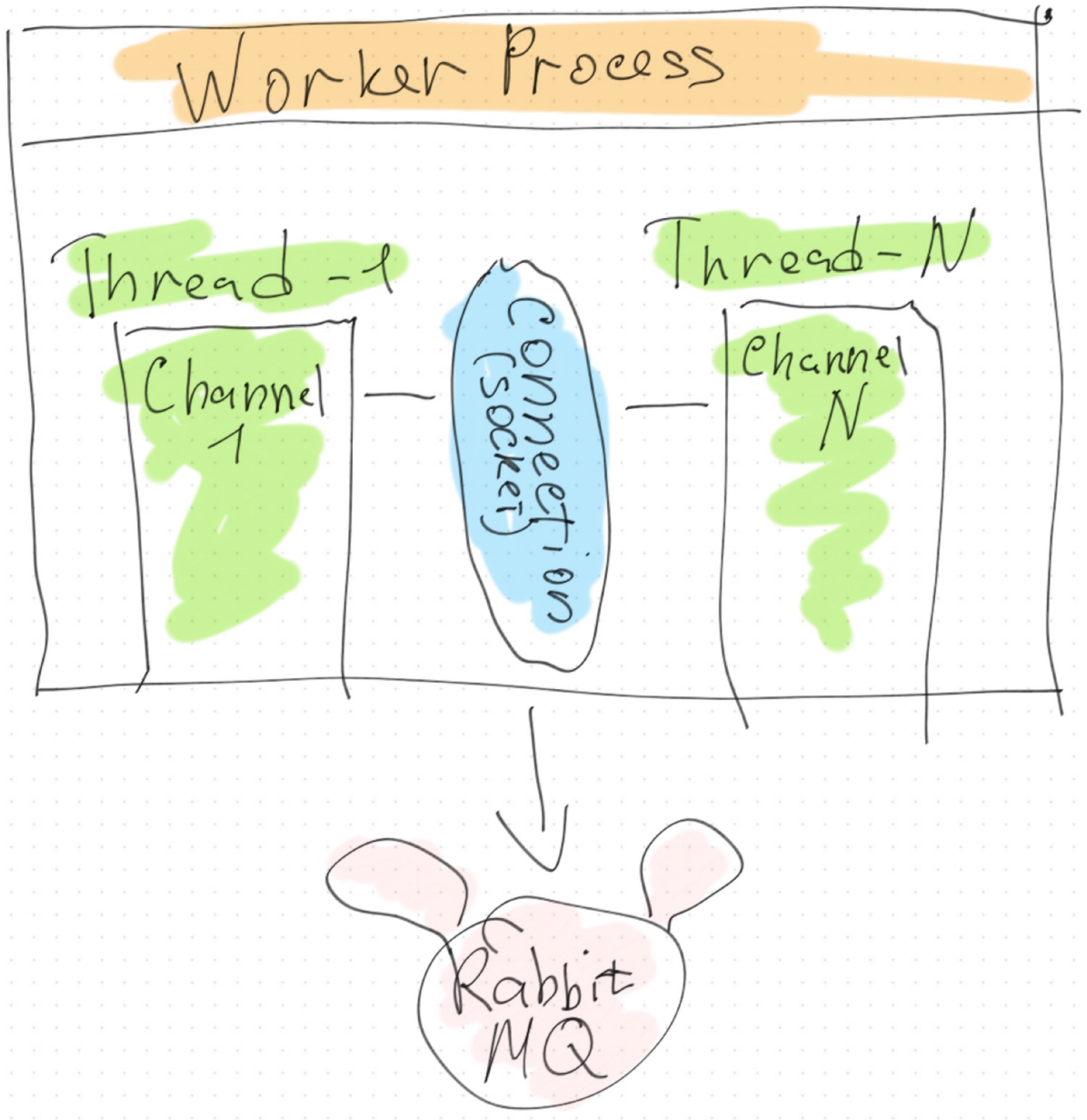
Dramatiq internals



Dramatiq internals



Dramatiq internals



Pika

Frequently Asked Questions

- Is Pika thread safe?

  Pika does not have any notion of threading in the code. If you want to use Pika with threading, make sure you have a Pika connection per thread, created in that thread. It is not safe to share one Pika connection across threads, with one exception: you may call the connection method *add_callback_threadsafe* from another thread to schedule a callback within an active pika connection. See connection adapter for more details


никакого праздника!

Тебя не обвинят в ошибке, если это официальная рекомендация

Dramatiq uses more RabbitMQ connections than necessary #649

Open **WillDaSilva** opened this issue 2 days ago · 1 comment

WillDaSilva commented 2 days ago — Contributor

We use RabbitMQ as our Dramatiq broker, and we're always dangerously close to the connection limit (which is 500 in our case). This is because Dramatiq creates a connection in each worker process for each queue we have, and we have a queue per Dramatiq task.

We think Dramatiq can be modified to use fewer connections. According to CloudAMQP's RabbitMQ best practices, you should generally limit connections to one connection per process, and use channels within that connection for per-thread use. Instead of doing this, Dramatiq creates a new connection with a single channel for each RabbitMQ consumer it creates, and it creates a consumer for each queue per process, when it could probably be made to just use one connection which each consumer within a process shares, and then each consumer can just create a channel for itself.

What OS are you using?

Debian 12 (via the official Python Docker image)

What version of Dramatiq are you using?

v1.17.0

Dramatiq uses more RabbitMQ connections than necessary #649

Open **WillDaSilva** opened this issue 2 days ago · 1 comment

WillDaSilva commented 2 days ago — Contributor

We use RabbitMQ as our Dramatiq broker, and we're always dangerously close to the connection limit (which
case). This is because Dramatiq creates a connection in each worker process for each queue we have, and
per Dramatiq task.

We think Dramatiq can be modified to use fewer connections. According to CloudAMQP's RabbitMQ b
should generally limit connections to one connection per process, and use channels within that co
use. Instead of doing this, Dramatiq creates a new connection with a single channel for each Rabbit
and it creates a consumer for each queue per process, when it could probably be made to just use or
each consumer within a process shares, and then each consumer can just create a channel for itself.

What OS are you using?

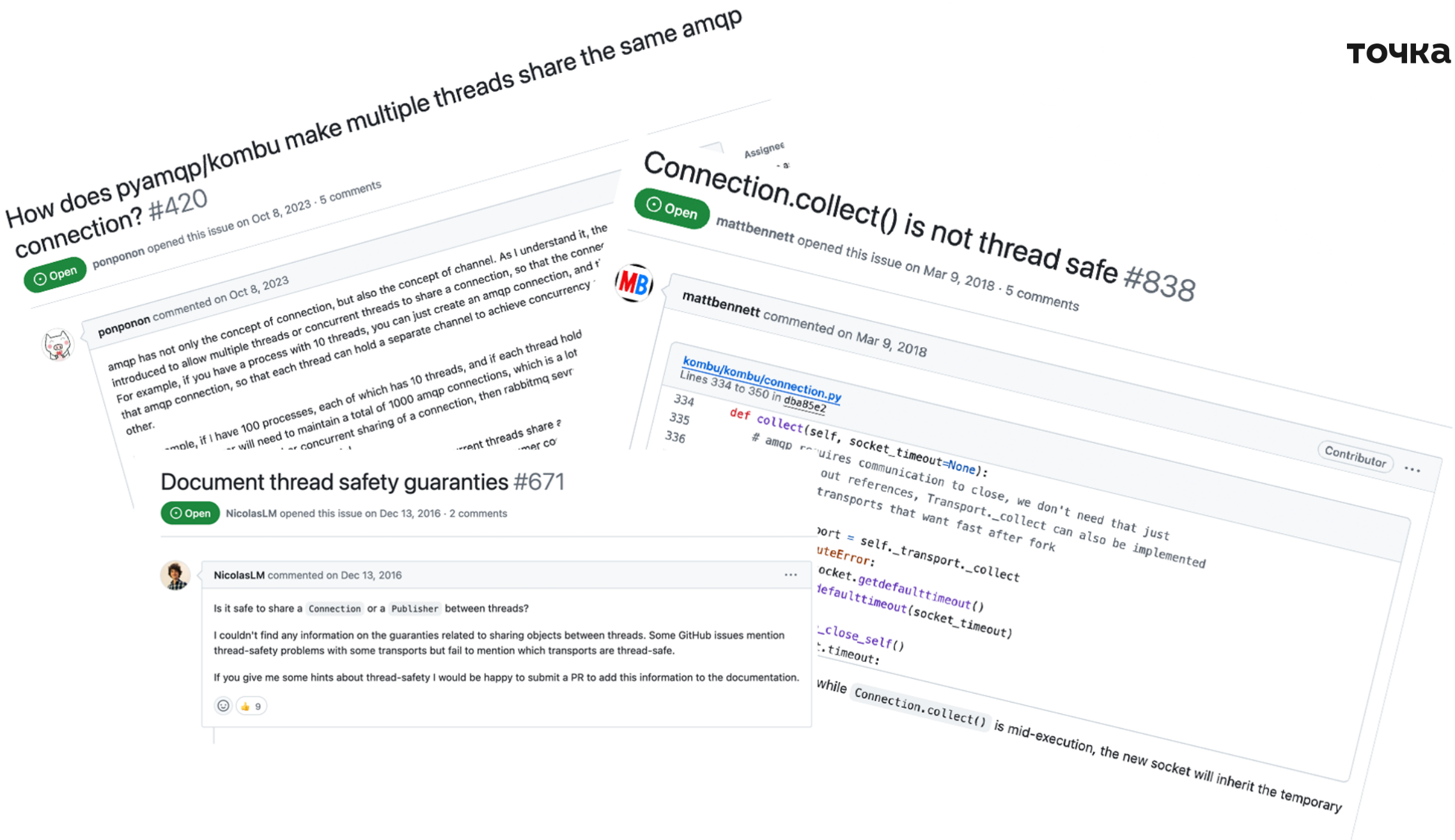
Debian 12 (via the official Python Docker image)

What version of Dramatiq are you using?

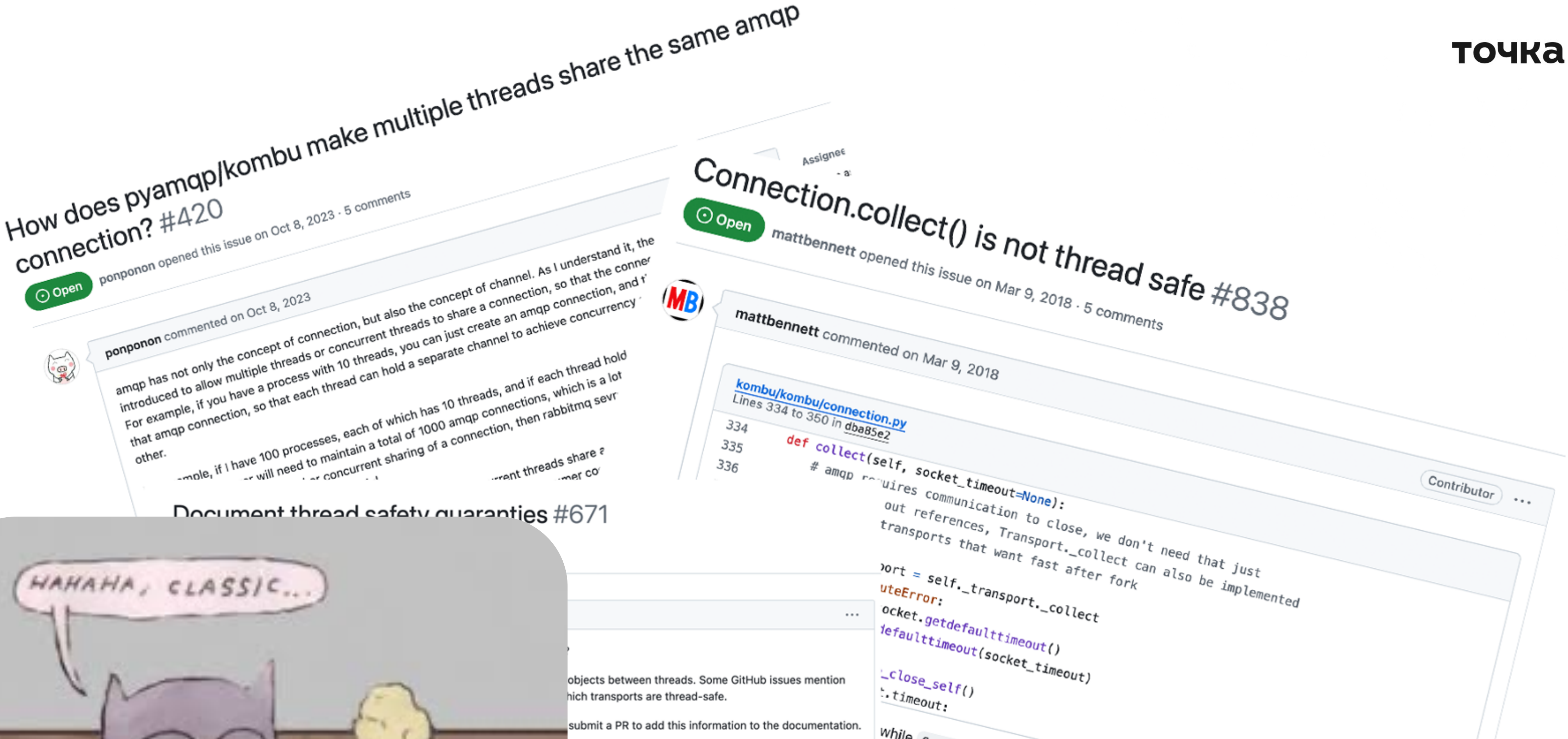
v1.17.0

Kombu

точка


How does pyamqp/kombu make multiple threads share the same amqp connection? #420
Open
ponponon opened this issue on Oct 8, 2023 · 5 comments
ponponon commented on Oct 8, 2023
amqp has not only the concept of connection, but also the concept of channel. As I understand it, the
introduced to allow multiple threads or concurrent threads to share a connection, so that the
For example, if you have a process with 10 threads, you can just create an amqp connection, and
that amqp connection, so that each thread can hold a separate channel to achieve concurrency
other.
Document thread safety guaranties #671
Open
NicolasLM opened this issue on Dec 13, 2016 · 2 comments
NicolasLM commented on Dec 13, 2016
Is it safe to share a Connection or a Publisher between threads?
I couldn't find any information on the guaranties related to sharing objects between threads. Some GitHub issues mention thread-safety problems with some transports but fail to mention which transports are thread-safe.
If you give me some hints about thread-safety I would be happy to submit a PR to add this information to the documentation.
9
Connection.collect() is not thread safe #838
Open
mattbennett opened this issue on Mar 9, 2018 · 5 comments
mattbennett commented on Mar 9, 2018
kombu/kombu/connection.py
Lines 334 to 350 in dba85e2
Contributor
def collect(self, socket_timeout=None):
while Connection.collect() is mid-execution, the new socket will inherit the temporary

точка



How does pyamqp/kombu make multiple threads share the same amqp connection? #420

Open ponponon opened this issue on Oct 8, 2023 · 5 comments

ponponon commented on Oct 8, 2023

amqp has not only the concept of connection, but also the concept of channel. As I understand it, the introduced to allow multiple threads or concurrent threads to share a connection, so that the conne For example, if you have a process with 10 threads, you can just create an amqp connection, and t that amqp connection, so that each thread can hold a separate channel to achieve concurrency other.

...mple, if I have 100 processes, each of which has 10 threads, and if each thread hold ... will need to maintain a total of 1000 amqp connections, which is a lot ... concurrent sharing of a connection, then rabbitmq serv ... rent threads share ...

Connection.collect() is not thread safe #838

Open mattbennett opened this issue on Mar 9, 2018 · 5 comments

mattbennett commented on Mar 9, 2018

Contributor

kombu/kombu/connection.py
Lines 334 to 350 in dba85e2

```
def collect(self, socket_timeout=None):
    # amqp requires communication to close, we don't need that just
    ...out references, Transport._collect can also be implemented
    ...transports that want fast after fork
    ...port = self._transport._collect
    ...uteError:
    ...ocket.getdefaulttimeout()
    ...defaulttimeout(socket_timeout)
    ..._close_self()
    ....timeout:
```

while `Connection.collect()` is mid-execution, the new socket will inherit the temporary

Document thread safety guarantees #671

...objects between threads. Some GitHub issues mention ...hich transports are thread-safe.

...submit a PR to add this information to the documentation.

точка

точка

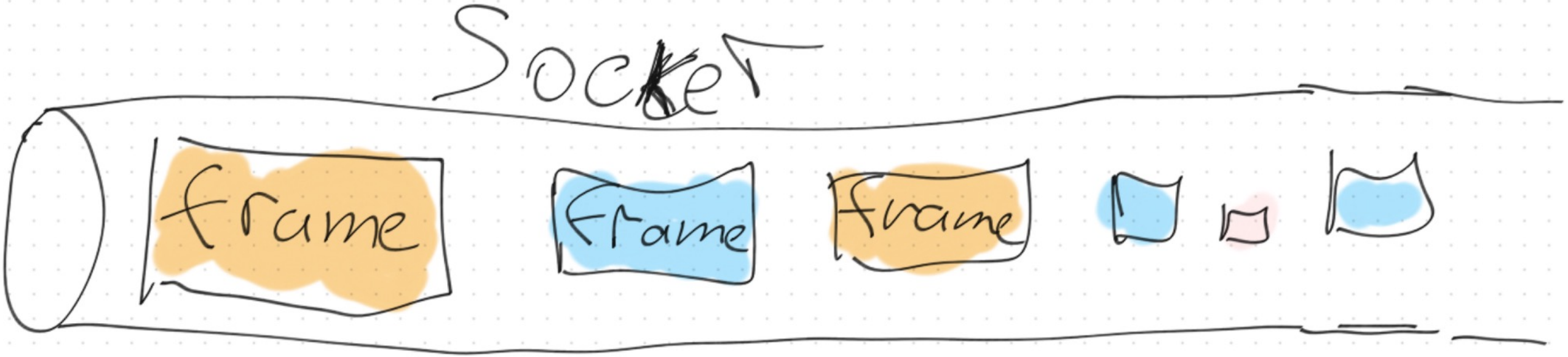
банк для предпринимателей
и предприятий

Приручение *Кролика*

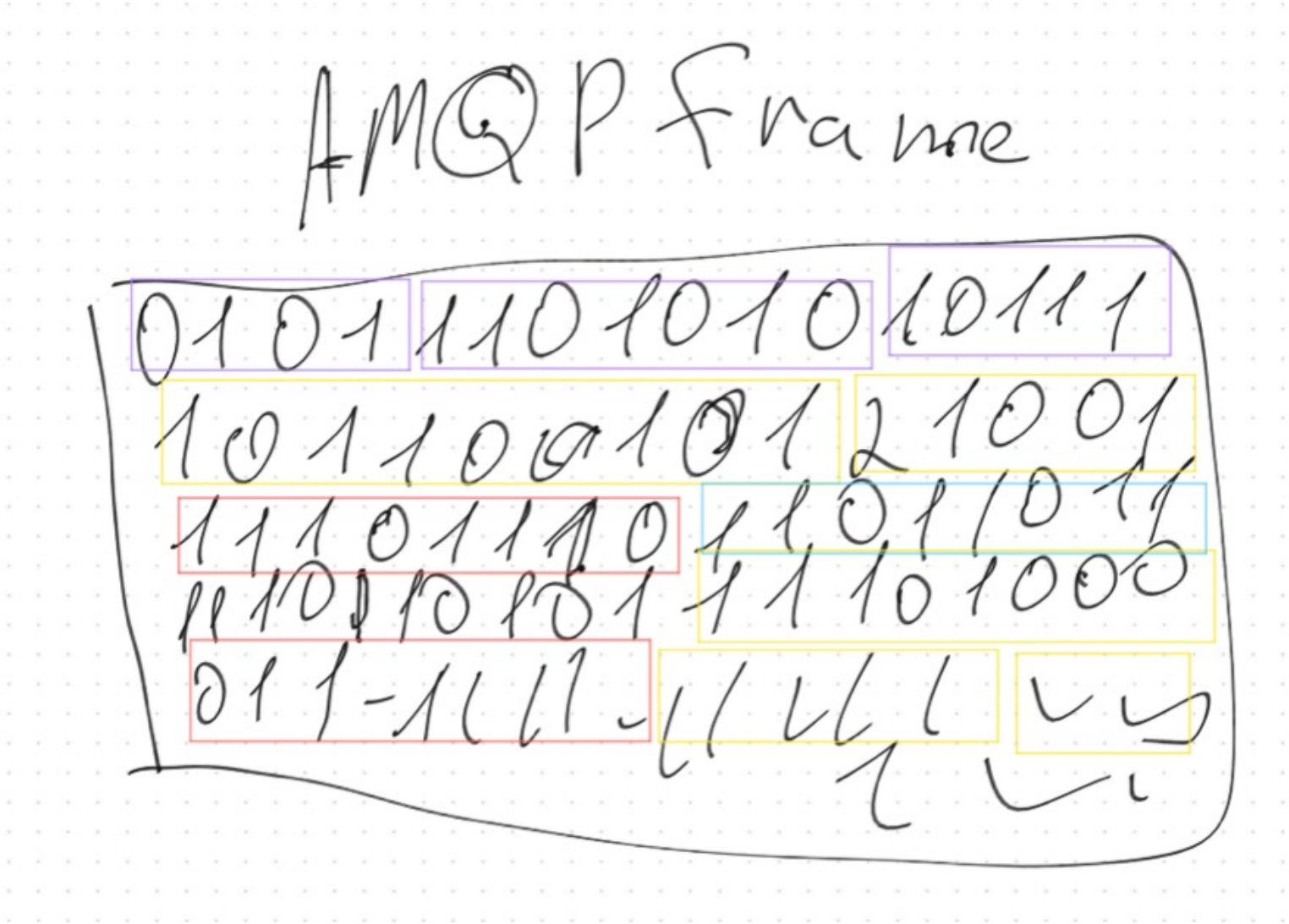

deeper

Socket

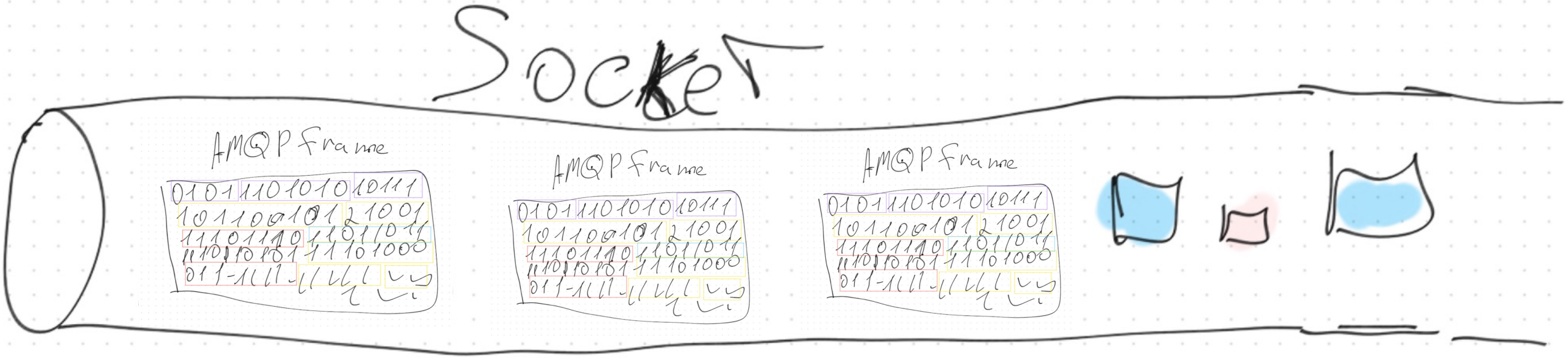
Приручение кролика: socket



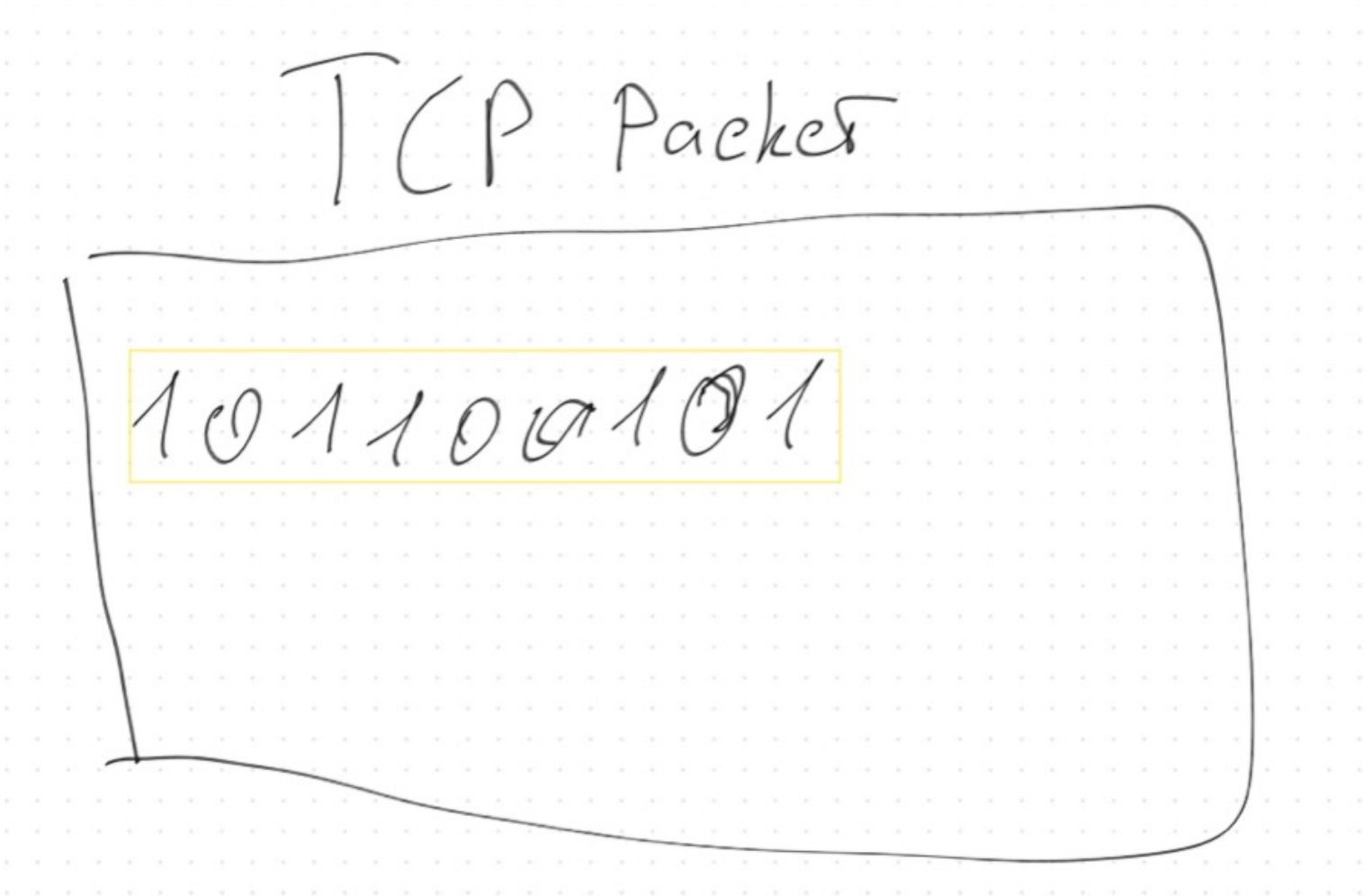
Приручение кролика: socket



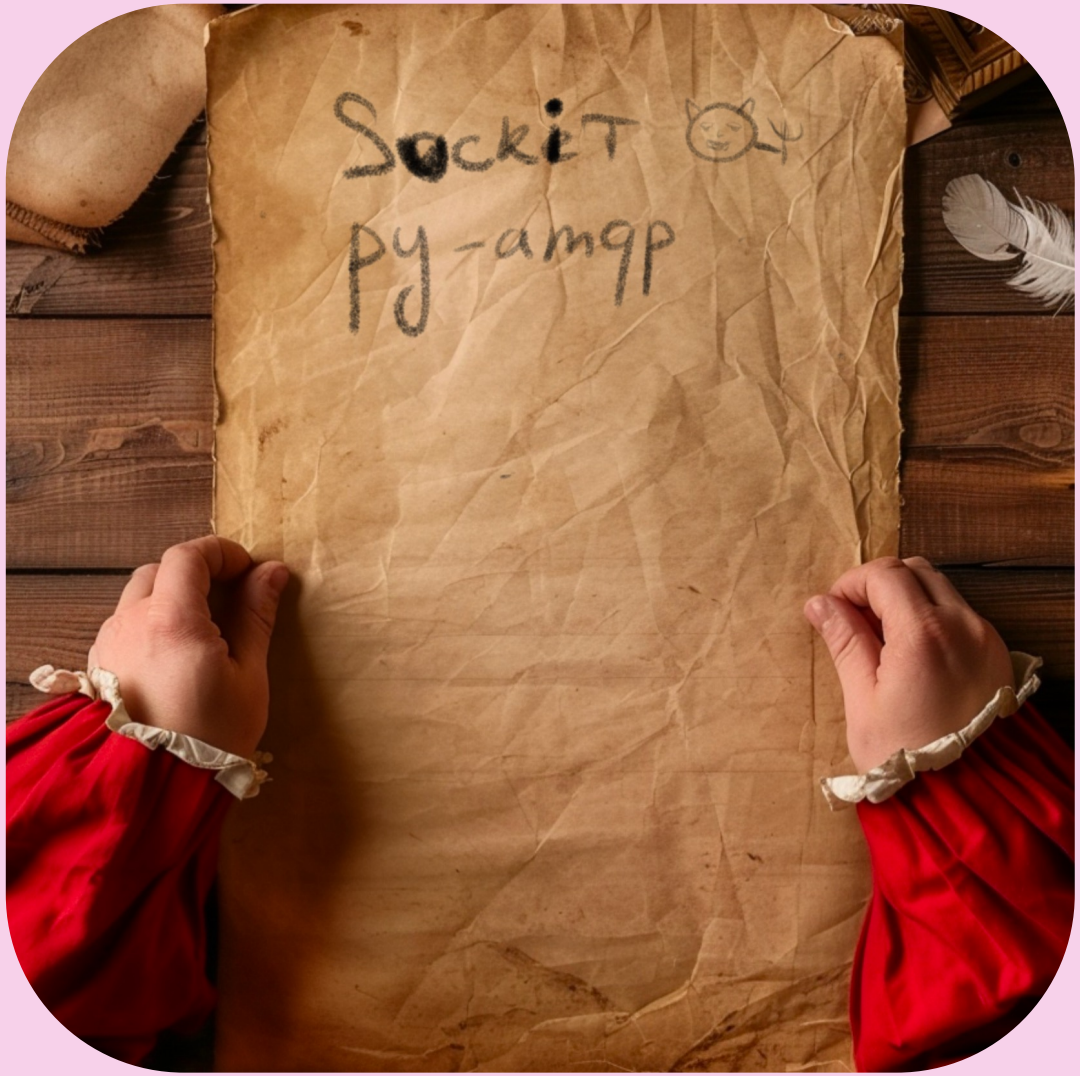
Приручение кролика: socket



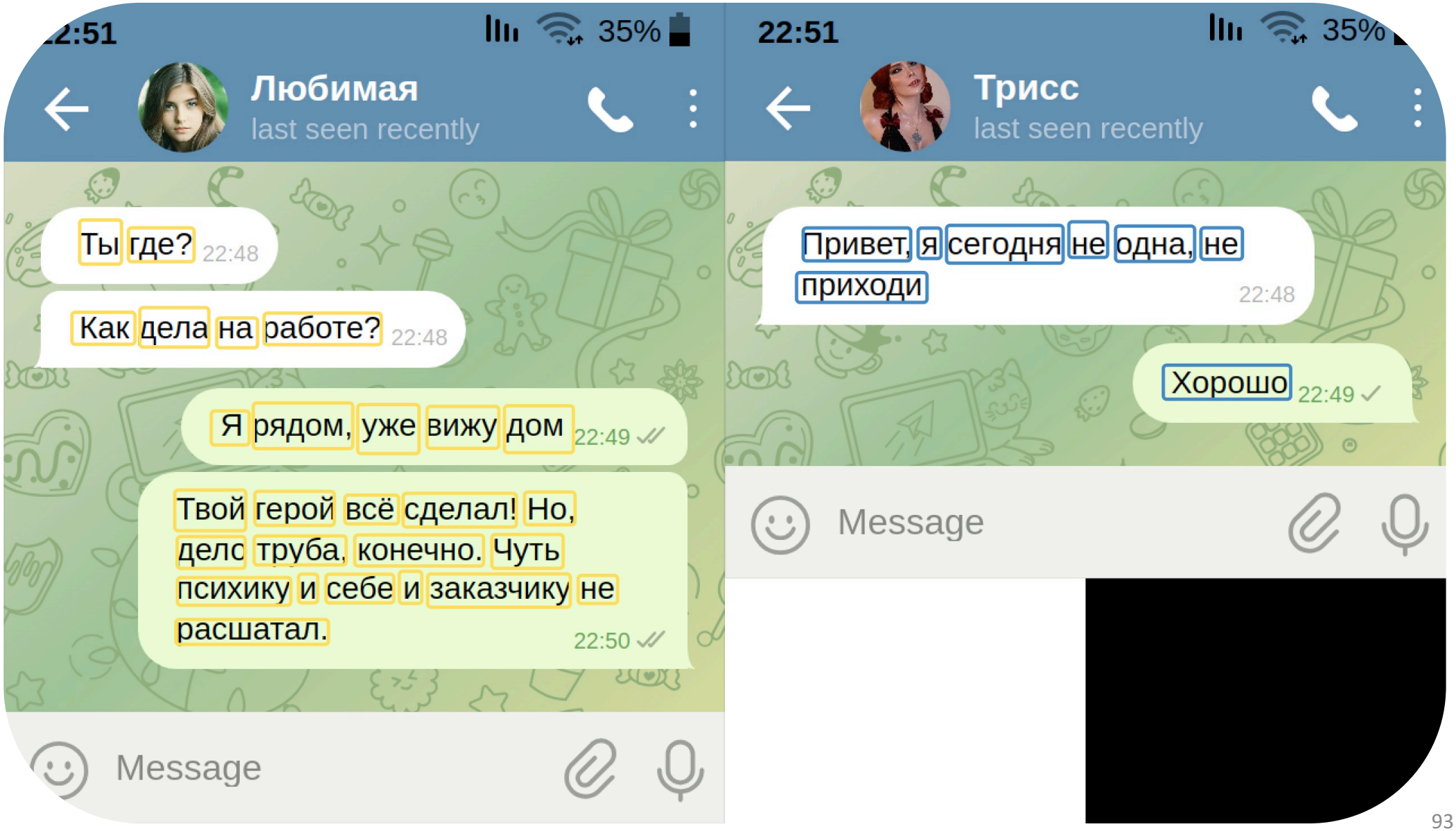
Приручение кролика: socket



py-amqp



Приручение кролика: py-amqp



Приручение кролика: py-amqp

amqp/transport.py

```python
    def read_frame(self, unpack=unpack):
        """Parse AMQP frame."""
        read = self._read
        read_frame_buffer = EMPTY_BUFFER
        try:
            frame_header = read(7, True)
            read_frame_buffer += frame_header
            frame_type, channel, size = unpack('>BHI', frame_header)
            # >I is an unsigned int, but the argument to sock.recv is signed,
            # so we know the size can be at most 2 * SIGNED_INT_MAX
            if size > SIGNED_INT_MAX:
                part1 = read(SIGNED_INT_MAX)
                ...
                try:
                  part2 = read(size - SIGNED_INT_MAX)
                except ...:
                  ...
                  raise
                payload = b''.join([part1, part2])
            else:
                payload = read(size)
            read_frame_buffer += payload
            frame_end = ord(read(1))
            ...
        except ...:
            ...
            raise
        # frame-end octet must contain '\xce' value
        if frame_end == 206:
            return frame_type, channel, payload
        else:
            raise UnexpectedFrame(
                f'Received frame_end {frame_end:#04x} while expecting 0xce')

```

amqp/transport.py

```python
def read_frame(self, unpack=unpack):
    """Parse AMQP frame."""
    read = self._read
    read_frame_buffer = EMPTY_BUFFER
    try:
        frame_header = read(7, True)
        read_frame_buffer += frame_header
        frame_type, channel, size = unpack('>BHI', frame_header)
        # >I is an unsigned int, but the argument to sock.recv is signed,
        # so we know the size can be at most 2 * SIGNED_INT_MAX
        if size > SIGNED_INT_MAX:
            part1 = read(SIGNED_INT_MAX)
            ...
            try:
              part2 = read(size - SIGNED_INT_MAX)
            except ...:
              ...
              raise
            payload = b''.join([part1, part2])
        else:
            payload = read(size)
        read_frame_buffer += payload
        frame_end = ord(read(1))
        ...
```

Приручение кролика: py-amqp

- Читаем **заголовок**

amqp/transport.py

```python
def read_frame(self, unpack=unpack):
    """Parse AMQP frame."""
    read = self._read
    read_frame_buffer = EMPTY_BUFFER
    try:
        frame_header = read(7, True)
        read_frame_buffer += frame_header
        frame_type, channel, size = unpack('>BHI', frame_header)
        # >I is an unsigned int, but the argument to sock.recv is signed,
        # so we know the size can be at most 2 * SIGNED_INT_MAX
        if size > SIGNED_INT_MAX:
            part1 = read(SIGNED_INT_MAX)
            ...
            try:
              part2 = read(size - SIGNED_INT_MAX)
            except ...:
              ...
              raise
            payload = b''.join([part1, part2])
        else:
            payload = read(size)
        read_frame_buffer += payload
        frame_end = ord(read(1))
        ...
    except ...:
        ...
        raise
    # frame-end octet must contain '\xce' value
    if frame_end == 206:
        return frame_type, channel, payload
    else:
        raise UnexpectedFrame(
            f'Received frame_end {frame_end:#04x} while expecting 0xce')
```

Приручение кролика: py-amqp

- Читаем **заголовок**
- Понимаем **что** перед нами

amqp/transport.py

```python
    def read_frame(self, unpack=unpack):
        """Parse AMQP frame."""
        read = self._read
        read_frame_buffer = EMPTY_BUFFER
        try:
            frame_header = read(7, True)
            read_frame_buffer += frame_header
            frame_type, channel, size = unpack('>BHI', frame_header)
            # >I is an unsigned int, but the argument to sock.recv is signed,
            # so we know the size can be at most 2 * SIGNED_INT_MAX
            if size > SIGNED_INT_MAX:
                part1 = read(SIGNED_INT_MAX)
                ...
                try:
                  part2 = read(size - SIGNED_INT_MAX)
                except ...:
                  ...
                  raise
                payload = b''.join([part1, part2])
            else:
                payload = read(size)
            read_frame_buffer += payload
            frame_end = ord(read(1))
            ...
        except ...:
            ...
            raise
        # frame-end octet must contain '\xce' value
        if frame_end == 206:
            return frame_type, channel, payload
        else:
            raise UnexpectedFrame(
                f'Received frame_end {frame_end:#04x} while expecting 0xce')

```

Приручение кролика: py-amqp

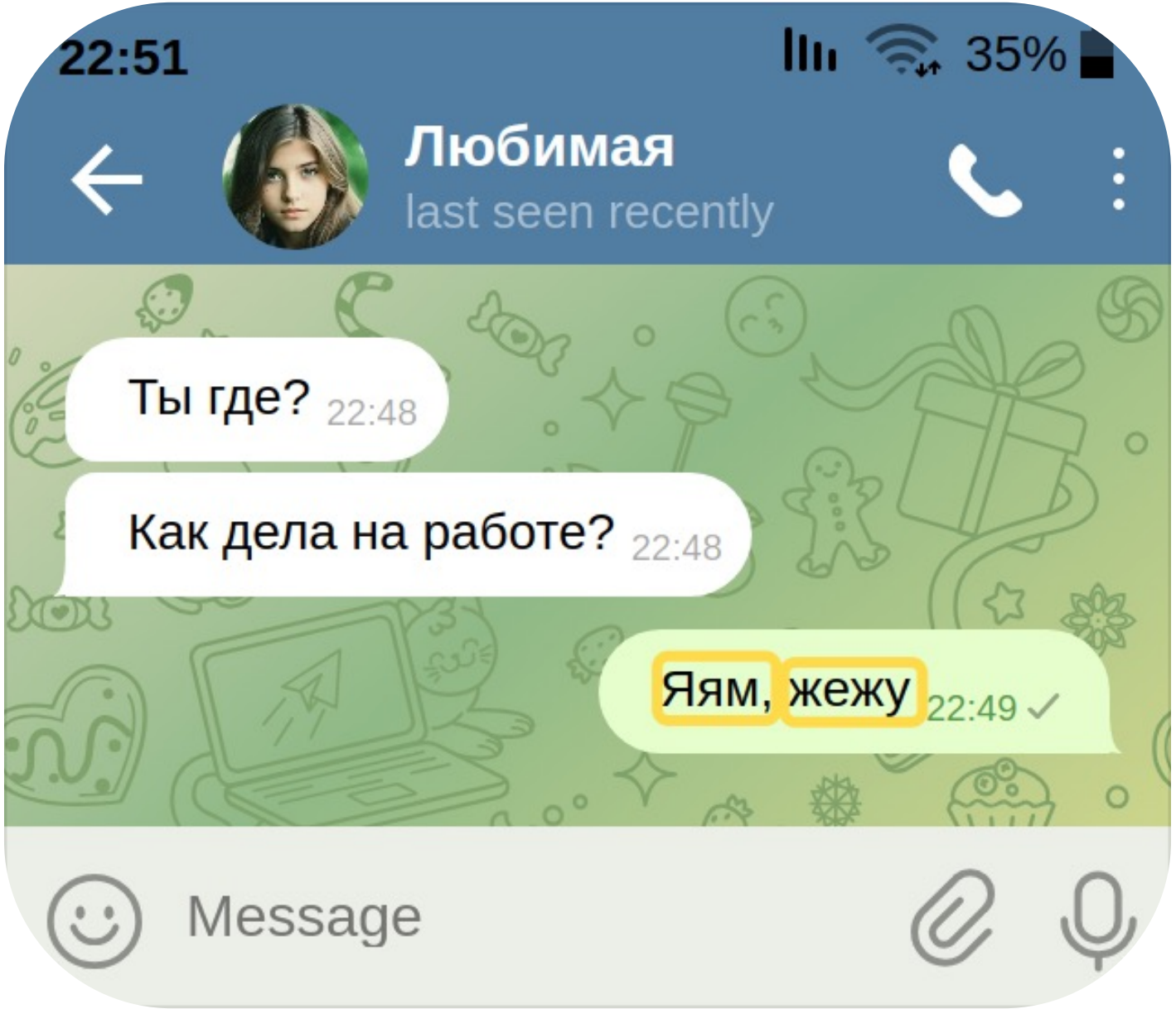
- Читаем **заголовок**
- Понимаем **что** перед нами
- Читаем **остальное**

amqp/transport.py

```
    def read_frame(self, unpack=unpack):
        """Parse AMQP frame."""
        read = self._read
        read_frame_buffer = EMPTY_BUFFER
        try:
            frame_header = read(7, True)
            read_frame_buffer += frame_header
            frame_type, channel, size = unpack('>BHI', frame_header)
            # >I is an unsigned int, but the argument to sock.recv is signed,
            # so we know the size can be at most 2 * SIGNED_INT_MAX
            if size > SIGNED_INT_MAX:
                part1 = read(SIGNED_INT_MAX)
                ...
                try:
                  part2 = read(size - SIGNED_INT_MAX)
                except ...:
                  ...
                  raise
                payload = b''.join([part1, part2])
            else:
                payload = read(size)
            read_frame_buffer += payload
            frame_end = ord(read(1))
            ...
        except ...:
            ...
            raise
        # frame-end octet must contain '\xce' value
        if frame_end == 206:
            return frame_type, channel, payload
        else:
            raise UnexpectedFrame(
                f'Received frame_end {frame_end:#04x} while expecting 0xce')

```

ThreadSafe?

Приручение кролика: py-amqp



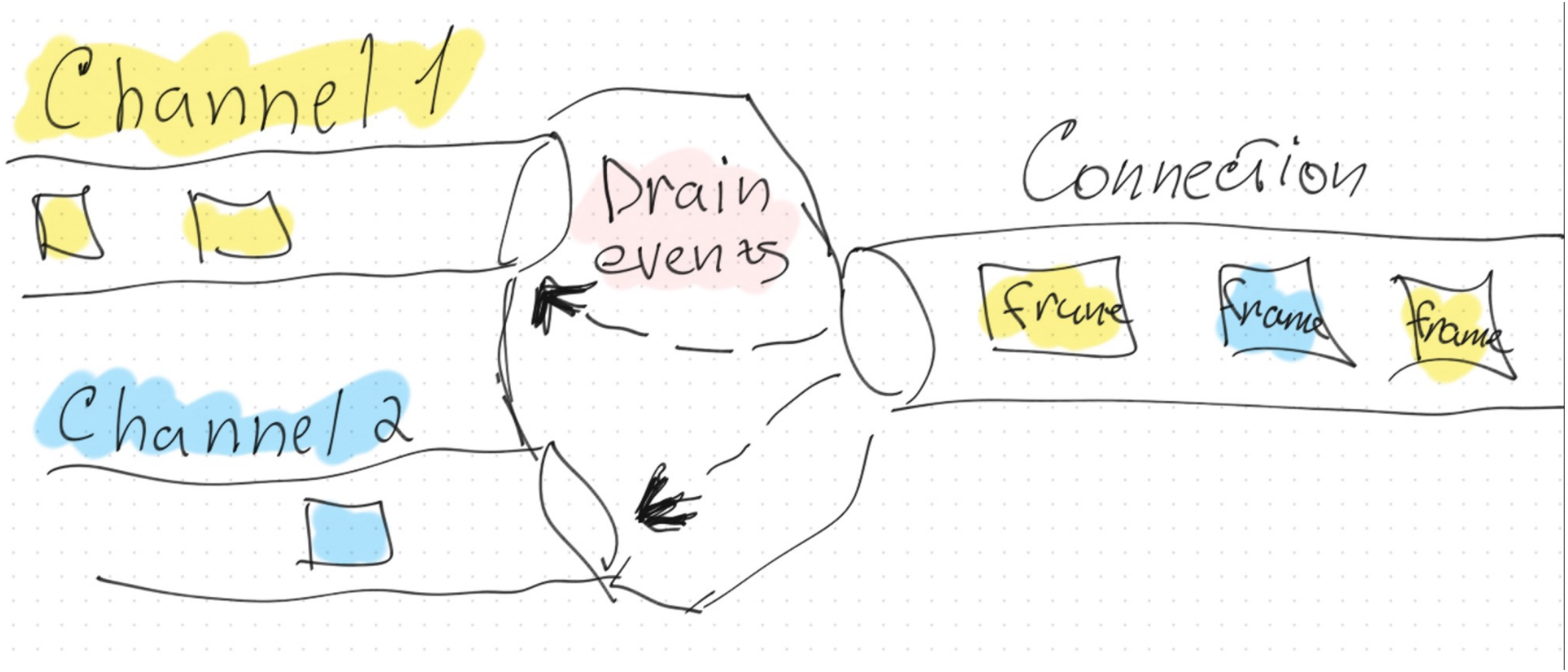
Problems:

- read_frame

py-amqp

Приручение кролика: py-amqp

Приручение кролика: py-amqp



Приручение кролика: py-amqp: Connection

amqp/connection.py

```
1     def drain_events(self, timeout=None):
2         # read until message is ready
3         while not self.blocking_read(timeout):
4             pass
5
6     def blocking_read(self, timeout=None):
7         with self.transport.having_timeout(timeout):
8             frame = self.transport.read_frame()
9         return self.on_inbound_frame(frame)
10
11    def on_inbound_method(self, channel_id, method_sig, payload, content):
12        if self.channels is None:
13            raise RecoverableConnectionError('Connection already closed')
14
15        return self.channels[channel_id].dispatch_method(
16            method_sig, payload, content,
17        )
18
```

Приручение кролика: py-amqp: AbstractChannel

- send_method

amqp/abstract_channel.py

```python
    def send_method(self, ...):
        ...
        args = dumps(format, args) if format else ''
        conn.frame_writer(1, self.channel_id, sig, args, content)
        ...

        if wait:
            return self.wait(wait, returns_tuple=returns_tuple)
        return ...


    def wait(self, method, callback=None, timeout=None, returns_tuple=False):
        p = ensure_promise(callback)
        pending = self._pending
        prev_p = []
        if not isinstance(method, list):
            method = [method]

        for m in method:
            prev_p.append(pending.get(m))
            pending[m] = p

        try:
            while not p.ready:
                self.connection.drain_events(timeout=timeout)

            if p.value:
                args, kwargs = p.value
                args = args[1:]  # We are not returning method back
                return args if returns_tuple else (args and args[0])
        finally:
            ...

```

Приручение кролика: py-amqp: AbstractChannel

- send_method
  - отправил запрос
  - ждет ответ (self.wait)

amqp/abstract_channel.py

```python
    def send_method(self, ...):
        ...
        args = dumps(format, args) if format else ''
        conn.frame_writer(1, self.channel_id, sig, args, content)
        ...

        if wait:
            return self.wait(wait, returns_tuple=returns_tuple)
        return ...


    def wait(self, method, callback=None, timeout=None, returns_tuple=False):
        p = ensure_promise(callback)
        pending = self._pending
        prev_p = []
        if not isinstance(method, list):
            method = [method]

        for m in method:
            prev_p.append(pending.get(m))
            pending[m] = p

        try:
            while not p.ready:
                self.connection.drain_events(timeout=timeout)

            if p.value:
                args, kwargs = p.value
                args = args[1:]  # We are not returning method back
                return args if returns_tuple else (args and args[0])
        finally:
            ...

```

Приручение кролика: py-amqp: AbstractChannel

- send_method
  - отправил запрос
  - ждет ответ (self.wait)
- wait
  - drain_events
  - проверка промиса

amqp/abstract_channel.py

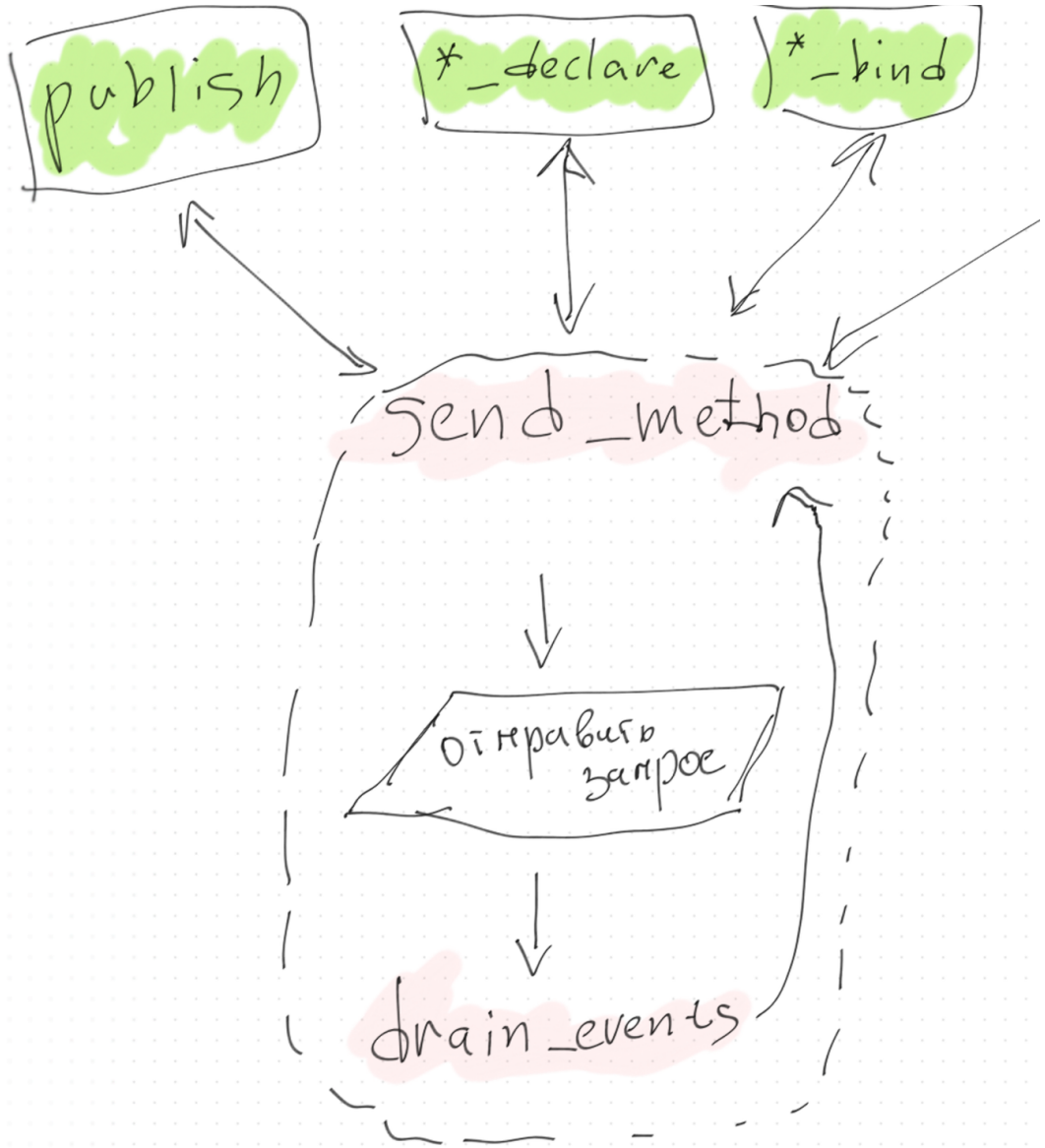
```
1    def send_method(self, ...):
2        ...
3        args = dumps(format, args) if format else ''
4        conn.frame_writer(1, self.channel_id, sig, args, content)
5        ...
6
7        if wait:
8            return self.wait(wait, returns_tuple=returns_tuple)
9        return ...
10
11
12   def wait(self, method, callback=None, timeout=None, returns_tuple=False):
13       p = ensure_promise(callback)
14       pending = self._pending
15       prev_p = []
16       if not isinstance(method, list):
17           method = [method]
18
19       for m in method:
20           prev_p.append(pending.get(m))
21           pending[m] = p
22
23       try:
24           while not p.ready:
25               self.connection.drain_events(timeout=timeout)
26
27           if p.value:
28               args, kwargs = p.value
29               args = args[1:]  # We are not returning method back
30               return args if returns_tuple else (args and args[0])
31       finally:
32           ...
33
```

Приручение кролика: py-amqp

Method send_method (AbstractChannel in amqp.abstr

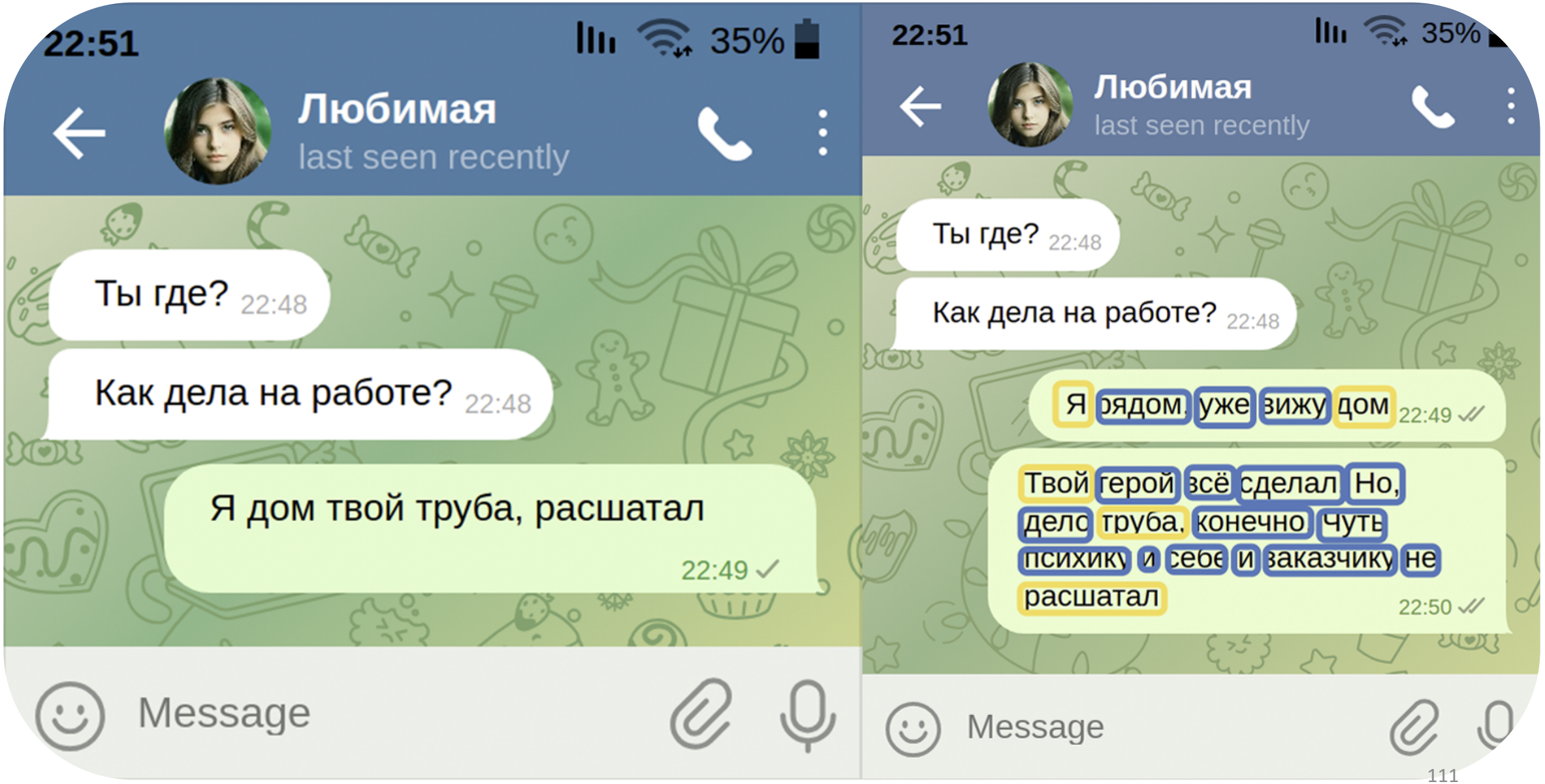
Project Files

channel.py	235	return self.**send_method**(
channel.py	290	self.**send_method**(spec.Cha
channel.py	357	return self.**send_method**(
channel.py	424	return self.**send_method**(sp
channel.py	448	return self.**send_method**(
channel.py	624	self.**send_method**(
channel.py	674	return self.**send_method**(
channel.py	751	return self.**send_method**(
channel.py	808	return self.**send_method**(
channel.py	939	return self.**send_method**(
channel.py	996	return self.**send_method**(
channel.py	1156	self.**send_method**(
channel.py	1235	return self.**send_method**(
channel.py	1296	return self.**send_method**(
channel.py	1407	return self.**send_method**(
channel.py	1452	return self.**send_method**(
channel.py	1574	p = self.**send_method**(
channel.py	1668	ret = self.**send_method**(
channel.py	1797	return self.**send_method**(
channel.py	1894	return self.**send_method**(
channel.py	1928	return self.**send_method**(sp
channel.py	1931	return self.**send_method**(sp
channel.py	2002	return self.**send_method**(
channel.py	2083	return self.**send_method**(sp
channel.py	2092	return self.**send_method**(sp
channel.py	2101	return self.**send_method**(sp
channel.py	2116	return self.**send_method**(
connection.py	416	self.**send_method**(
connection.py	442	self.**send_method**(
connection.py	449	self.**send_method**(

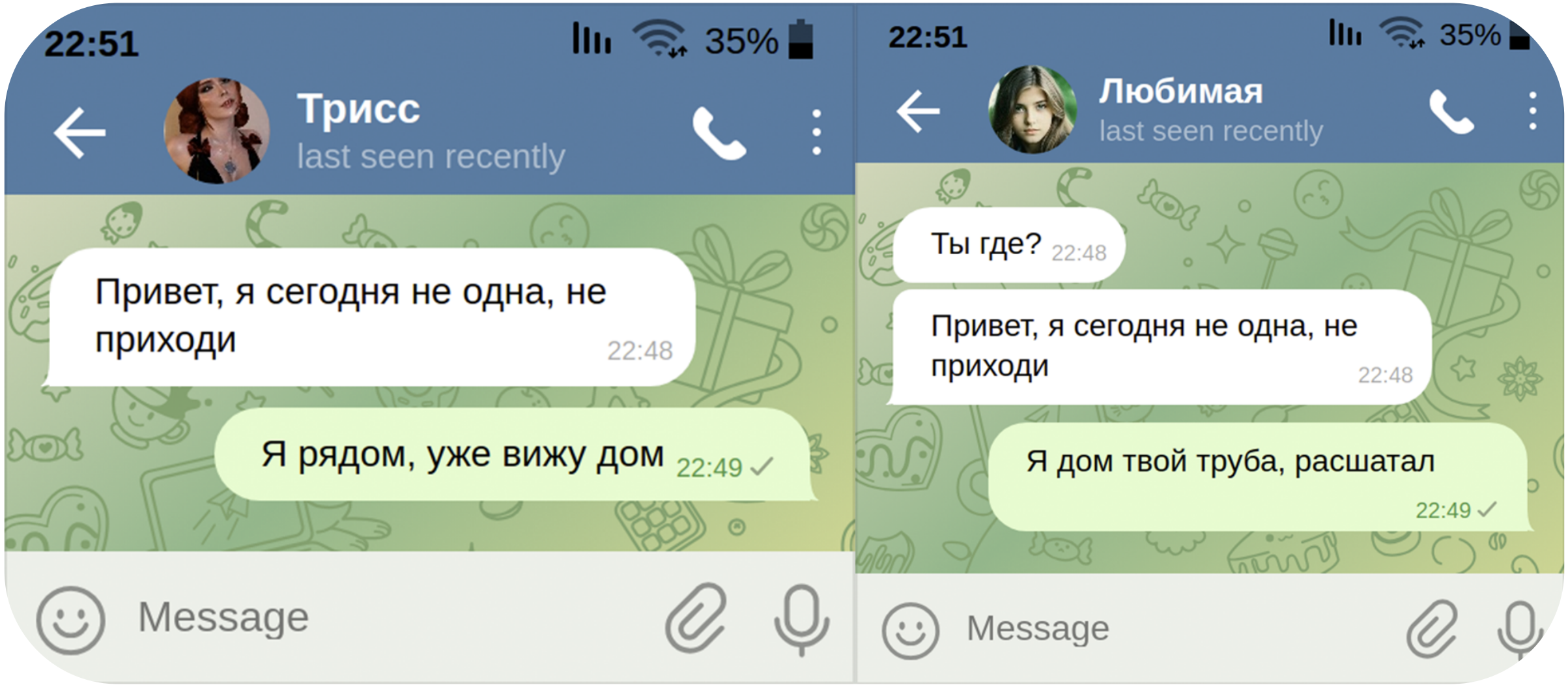


ThreadSafe?

Приручение кролика: py-amqp

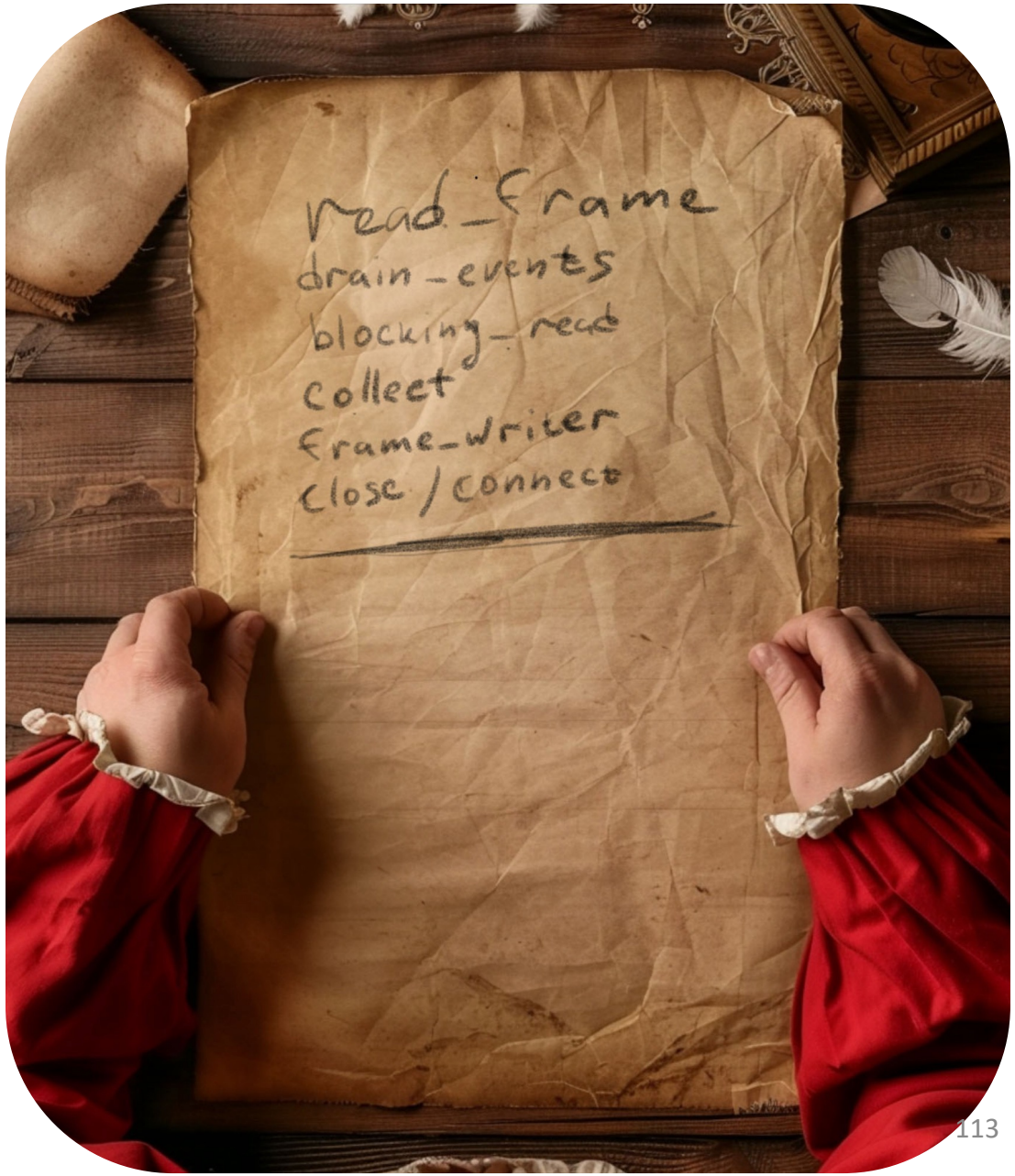


Приручение кролика: py-amqp

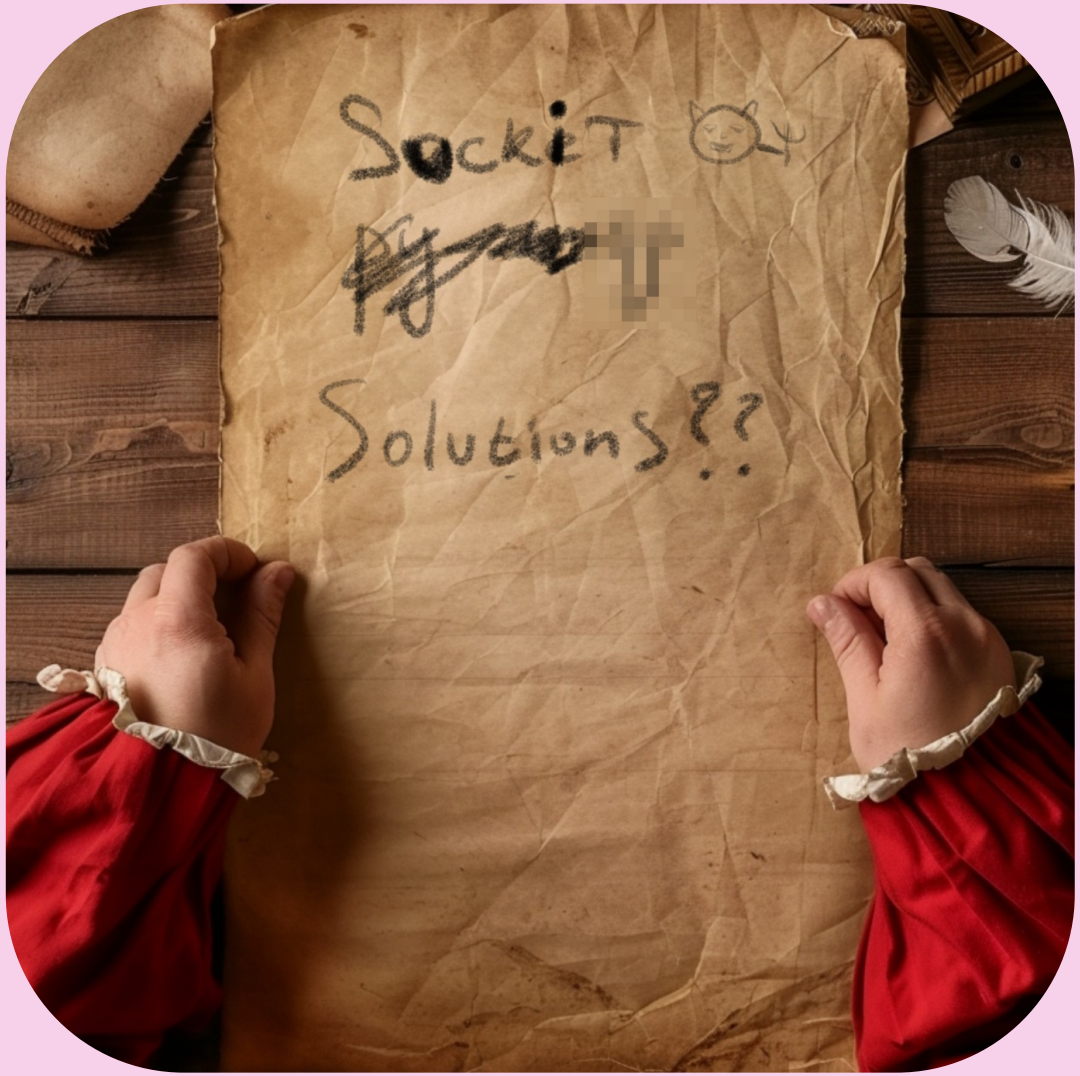


Problems:

- read_frame
- drain_events
- blocking_read
- collect
- frame_wirter
- close
- connect



Решение



Варианты решений:

Варианты решений:

1. кто-то уже сделал всё за нас

Варианты решений:

1. кто-то уже сделал всё за нас
2. смена стека

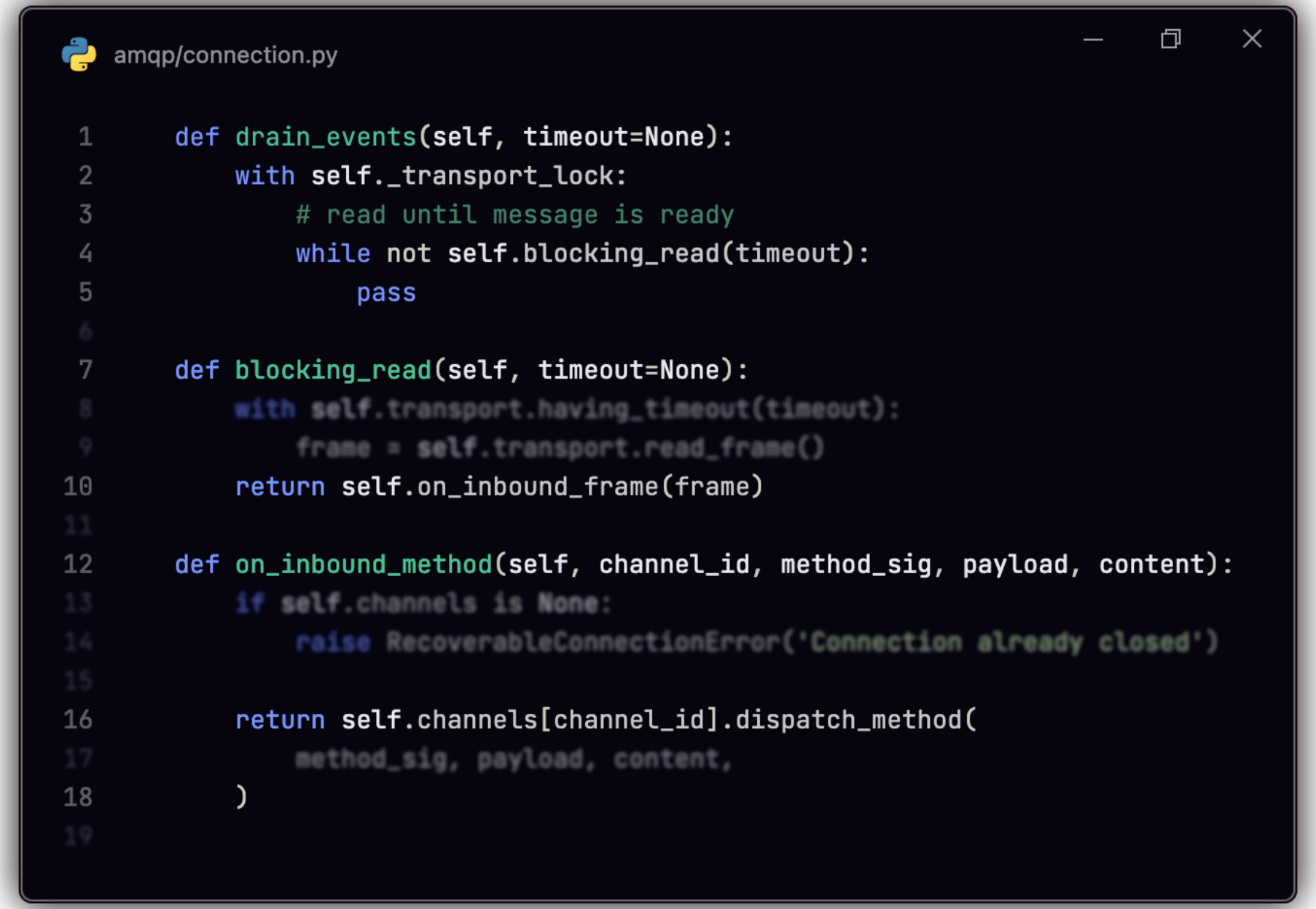
Варианты решений:

1. кто-то уже сделал всё за нас
2. смена стека (фастстрим то уже есть)
3. сделать самим

Locks


locks.py
1 lock = threading.Lock()
2
3 def critical_section():
4 with lock:
5 # now safe
6 ...


amqp/connection.py
1 def drain_events(self, timeout=None):
2 with self._transport_lock:
3 # read until message is ready
4 while not self.blocking_read(timeout):
5 pass
6


amqp/connection.py
def drain_events(self, timeout=None):
with self._transport_lock:
# read until message is ready
while not self.blocking_read(timeout):
pass
def blocking_read(self, timeout=None):
with self.transport.having_timeout(timeout):
frame = self.transport.read_frame()
return self.on_inbound_frame(frame)
def on_inbound_method(self, channel_id, method_sig, payload, content):
if self.channels is None:
raise RecoverableConnectionError('Connection already closed')
return self.channels[channel_id].dispatch_method(
method_sig, payload, content,
)

amqp/connection.py

```
    def on_inbound_method(self, channel_id, method_sig, payload, content):
        ...
        # collect all frames to late dispatch (after drain)
        self.channel_frame_buff[channel_id].append(
            (method_sig, payload, content)
        )

```


amqp/connection.py
1 def drain_events(self, timeout=None):
2 with self._transport_lock:
3 while not self.blocking_read(timeout):
4 pass
5

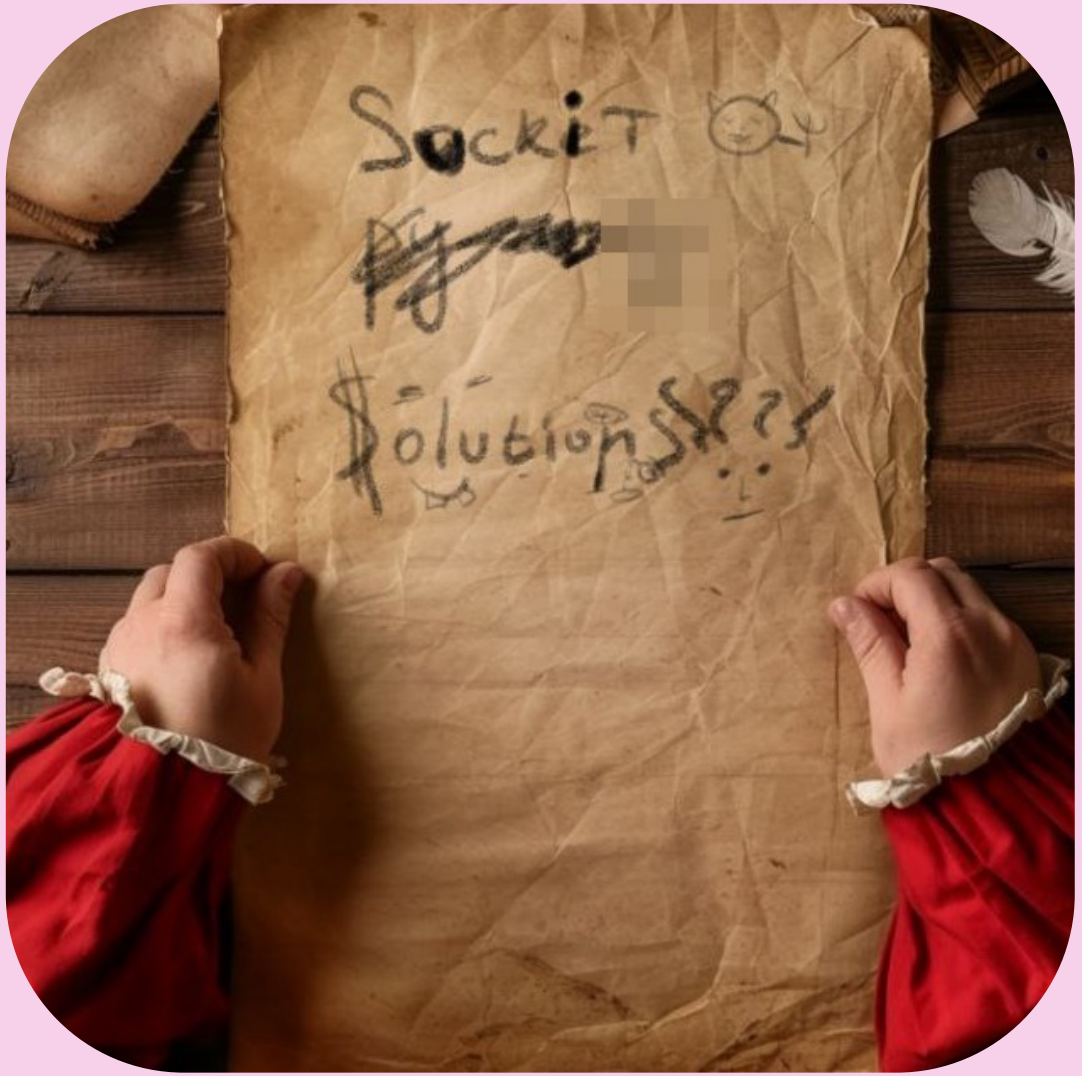
amqp/connection.py

```python
    _drain_cond: threading.Condition

    def drain_events(self, timeout=None):
      start_drain = False
      with self._drain_cond:
        if not self._drain_started:
            self._drain_started = True
            start_drain = True

      if start_drain:
        # сюда зайдет один поток
        with self._transport_lock:
          while not self.blocking_read(timeout):
            pass
        with self._drain_cond:
            self._drain_started = False
            self._drain_cond.notify_all()
      else:
        # сюда все остальные
        with self._drain_cond:
            self._drain_cond.wait(timeout=timeout)

```

amqp/connection.py

```
    _drain_guard: DrainGuard

    def drain_events(self, timeout=None):
        # When all threads go here only one really drain events,
        # because this action independent of caller.
        # All events will be dispatched to their channels

        started = self._drain_guard.start_drain()

        if not started:
            self._drain_guard.wait_drain_finished()
        else:
            try:
                with self._transport_lock:
                    super().drain_events(timeout=timeout)

            finally:
                self._drain_guard.finish_drain()

        self._dispatch_channel_frames(self.CONNECTION_CHANNEL_ID)

        me = threading.get_ident()
        my_channels = self.channel_thread_bindings[me]
        for channel_id in my_channels:
            self._dispatch_channel_frames(channel_id)


```

Сработало?



Как понять:

Как понять:

1. Меньше соединений

Как понять:

1. Меньше соединений
2. Больше каналов

Как понять:

1. Меньше соединений
2. Больше каналов
3. Жизнь стала лучше

RabbitMQ™ RabbitMQ 3.11.17 Erlang 25.3

Overview | Connections | Channels | Exchanges | Queues | Admin

Page 1 of 1 - Filter: Regex ?

Overview			Details			Network	
Name	User name	State	SSL / TLS	Protocol	Channels	From client	To client
10.103.12.48:56710 customer-dramatiq-687465945-v7sng	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.48:56722 customer-dramatiq-687465945-v7sng	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.77:36386 dramatiq-6c84dc7b5c-lcbt9	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.77:48070 dramatiq-6c84dc7b5c-lcbt9	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.77:48942 dramatiq-6c84dc7b5c-lcbt9	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.77:51310 dramatiq-6c84dc7b5c-lcbt9	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.77:51314 dramatiq-6c84dc7b5c-lcbt9	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.13.134:48790 app-85bd49d8bc-djm99	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.13.56:41474 dramatiq-6c94dc7b5c-d68cx	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.13.56:41488 dramatiq-6c94dc7b5c-d68cx	npd	running	•	AMQP 0-9-1	1	34 B/s	81 B/s
10.103.13.56:50540 dramatiq-6c84dc7b5c-d68cx	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.13.56:52800 dramatiq-6c84dc7b5c-d68cx	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.13.56:55288 dramatiq-6c84dc7b5c-d68cx	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.168:42446 ?	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s
10.103.6.56:13154 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:14255 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:17165 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:31312 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:35091 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:45891 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:50842 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:52908 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:60215 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:7819 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.8.74:7094 ?	npd	running	•	AMQP 0-9-1	2	1.3 KiB/s	82 KiB/s
10.106.13.53:41308 app-85bd49d8bc-9njb5	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.13.61:35352 dramatiq-6c84dc7b5c-hvqz7	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.13.61:50684 dramatiq-6c84dc7b5c-hvqz7	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.13.61:53550 dramatiq-6c84dc7b5c-hvqz7	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.13.61:55040 dramatiq-6c84dc7b5c-hvqz7	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.30:60012 customer-dramatiq-687465945-sbfj8	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.30:60016 customer-dramatiq-687465945-sbfj8	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.50:33394 dramatiq-6c84dc7b5c-47x8j	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.50:39288 dramatiq-6c84dc7b5c-47x8j	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.50:50726 dramatiq-6c84dc7b5c-47x8j	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.50:53742 dramatiq-6c84dc7b5c-47x8j	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.50:53764 dramatiq-6c84dc7b5c-47x8j	npd	running	•	AMQP 0-9-1	1	9 B/s	310 B/s
10.106.3.192:10171 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:12598 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:19855 ?	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s
10.106.3.192:29465 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:36934 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:43232 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:46880 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:46939 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:4953 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:50102 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:8541 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.214:39098 ?	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s

RabbitMQ™ RabbitMQ 3.11.17 Erlang 25.3

Overview | Connections | Channels | Exchanges | Queues | Admin

Page 1 of 1 - Filter: Regex ?

Overview			Details			Network	
Name	User name	State	SSL / TLS	Protocol	Channels	From client	To client
10.103.12.48:56710 customer-dramatiq-687465945-v7sng	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.48:56722 customer-dramatiq-687465945-v7sng	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.77:36386 dramatiq-6c84dc7b5c-lcbt9	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.77:48070 dramatiq-6c84dc7b5c-lcbt9	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.77:48942 dramatiq-6c84dc7b5c-lcbt9	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.77:51310 dramatiq-6c84dc7b5c-lcbt9	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.12.77:51314 dramatiq-6c84dc7b5c-lcbt9	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.13.134:48790 app-85bd49d8bc-djm99	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.13.56:41474 dramatiq-6c84dc7b5c-d68cx	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.13.56:41488 dramatiq-6c84dc7b5c-d68cx	npd	running	•	AMQP 0-9-1	1	34 B/s	81 B/s
10.103.13.56:50540 dramatiq-6c84dc7b5c-d68cx	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.13.56:52800 dramatiq-6c84dc7b5c-d68cx	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.13.56:55288 dramatiq-6c84dc7b5c-d68cx	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.168:42446 ?	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s
10.103.6.56:13154 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:14255 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:17165 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:31312 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:35091 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:45891 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:50842 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:52908 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:60215 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.6.56:7819 dramatiq-scheduler-5fb969df6b-k5nzj	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.103.8.74:7094 ?	npd	running	•	AMQP 0-9-1	2	1.3 KiB/s	82 KiB/s
10.106.13.53:41308 app-85bd49d8bc-9njb5	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.13.61:35352 dramatiq-6c84dc7b5c-hvqz7	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.13.61:50684 dramatiq-6c84dc7b5c-hvqz7	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.13.61:53550 dramatiq-6c84dc7b5c-hvqz7	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.13.61:55040 dramatiq-6c84dc7b5c-hvqz7	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.30:60012 customer-dramatiq-687465945-sbfj8	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.30:60016 customer-dramatiq-687465945-sbfj8	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.50:33394 dramatiq-6c84dc7b5c-47x8j	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.50:39288 dramatiq-6c84dc7b5c-47x8j	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.50:50726 dramatiq-6c84dc7b5c-47x8j	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.50:53742 dramatiq-6c84dc7b5c-47x8j	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.14.50:53764 dramatiq-6c84dc7b5c-47x8j	npd	running	•	AMQP 0-9-1	1	9 B/s	310 B/s
10.106.3.192:10171 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:12598 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:19855 ?	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s
10.106.3.192:29465 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:36934 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:43232 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:46880 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:46939 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:4953 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:50102 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.192:8541 dramatiq-scheduler-5fb969df6b-t9tpv	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s
10.106.3.214:39098 ?	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s

RabbitMQ™ RabbitMQ 3.11.17 Erlang 25.3

Overview | Connections | Channels | Exchanges | Queues | Admin

Connections

All connections (8)

Pagination

Page 1 of 1 - Filter: Regex ?

Overview			Details			Network		+/-
Name	**User name**	**State**	**SSL / TLS**	**Protocol**	**Channels**	**From client**	**To client**	
10.106.70.45:16912 ?	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s	
10.106.70.53:29093 ?	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s	
10.106.70.61:5822 dramatiq-scheduler-6676d87468-8bmws	npd	running	•	AMQP 0-9-1	10	0 B/s	0 B/s	
10.106.72.135:42860 dramatiq-5f6f7f6b68-9qkct	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s	
10.106.72.135:60104 dramatiq-5f6f7f6b68-9qkct	npd	running	•	AMQP 0-9-1	1	0 B/s	0 B/s	
10.106.72.23:33082 dramatiq-5f6f7f6b68-tddcc	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s	
10.106.72.23:51438 dramatiq-5f6f7f6b68-tddcc	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s	
10.106.74.161:58748 customer-dramatiq-6dbf796d58-958b5	npd	running	•	AMQP 0-9-1	2	0 B/s	0 B/s	

Open-source

github: spumer/kombu-pyamqp-threadsafe

Key features:

- Thread-safe connection
- Channel pool support

github: spumer/dramatiq-kombu-broker

Key features:

- Connection pool support
- Channel pool support

Выводы

1. Open-Source это про **открытый код**, но **не про решение ваших проблем**.

2. **Не ждите** решения от других, решайте сами.

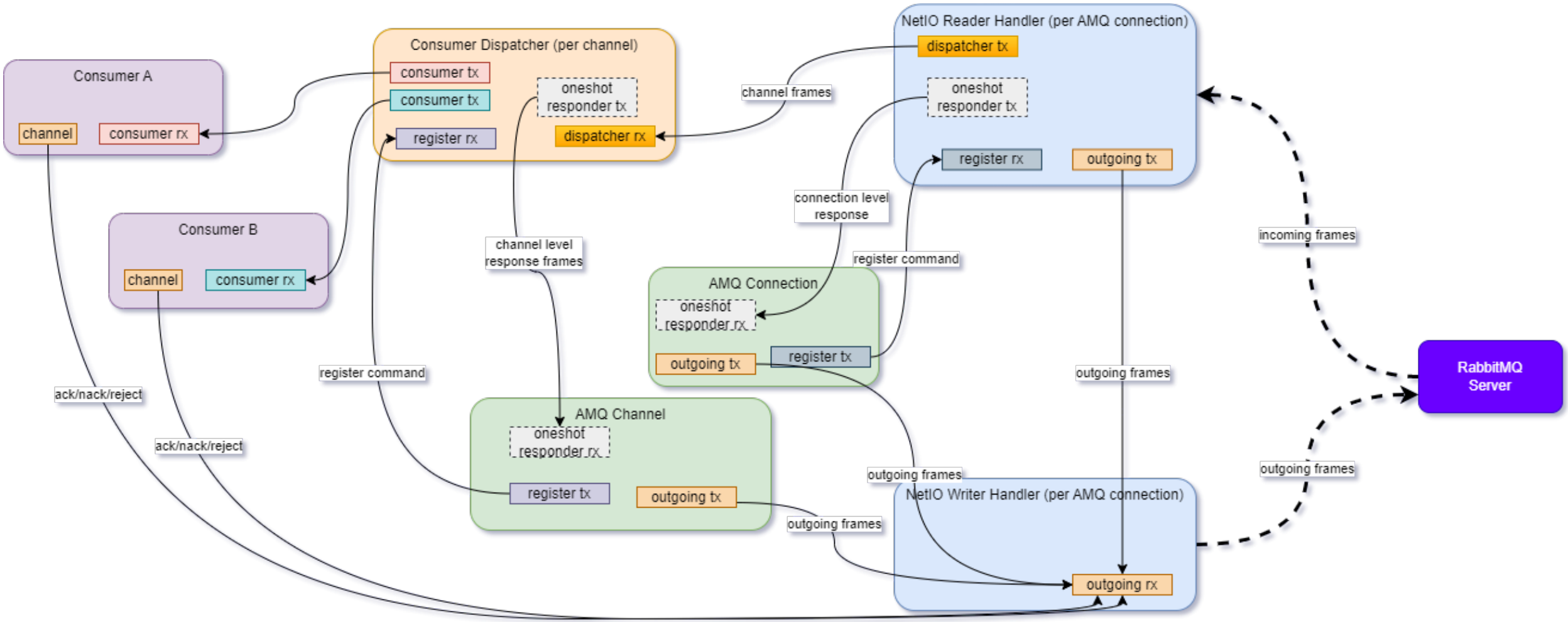
3. И конечно **поменьше велосипедов**, отдавайте ваш код на поддержку другим (PR welcome)

1. Open-Source это про **открытый код**, но **не про решение ваших проблем**.

2. **Не ждите** решения от других, решайте сами.

3. И конечно **поменьше велосипедов**, отдавайте ваш код на поддержку другим (PR welcome)

4. Это костыли, хочется уйти к Lock-free



https://github.com/gftea/amqprs


Присоединяйтесь!
Скоро ещё фич
подвезут!