

# WPROWADZENIE DO RABBITMQ I PHP-AMQLIB



## BARTOSZ ŁUCZAK

- ▶ W TSH od listopada 2020
- ▶ Zacząłem pracę jako programista PHP ok. 3.5 roku temu

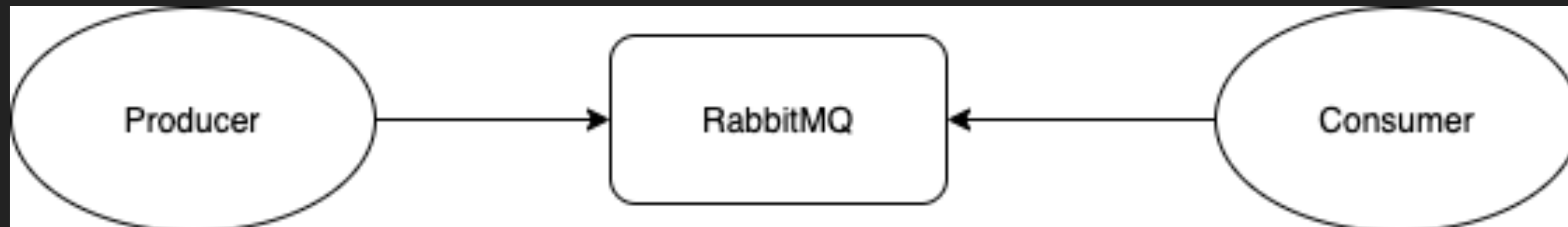
- ▶ Po co nam kolejki i czym jest RabbitMQ
- ▶ Zalety i przypadki użycia RabbitMQ
- ▶ RabbitMQ + php-amqplib
  - ▶ Wysyłka i odbiór wiadomości
  - ▶ QoS, zatwierdzanie i rekolejkowanie wiadomości
  - ▶ Centrale wiadomości (exchanges) i typy centrali - direct, fanout, headers, topic
  - ▶ Opóźnianie powrotu wiadomości do kolejki - Dead Letter Exchanges
- ▶ Gotowe pakiety do wykorzystania z Symfony i Laravelem

## PO CO NAM KOLEJKOWANIE

- ▶ Współpraca między serwisami/procesami, które mogą nie działać w tym samym czasie
- ▶ Może podnieść niezawodność budowanych systemów poprzez gwarancję przekazania wiadomości
- ▶ Może podnieść wydajność systemu wprowadzając przetwarzanie asynchroniczne

## CZYM JEST RABBITMQ

- ▶ Pośrednik wiadomości / System kolejkowania
- ▶ Wykorzystywany do wytwarzania systemów rozproszonych
- ▶ Wspiera wiele języków m.in. PHP, Java, .NET, Ruby, Python



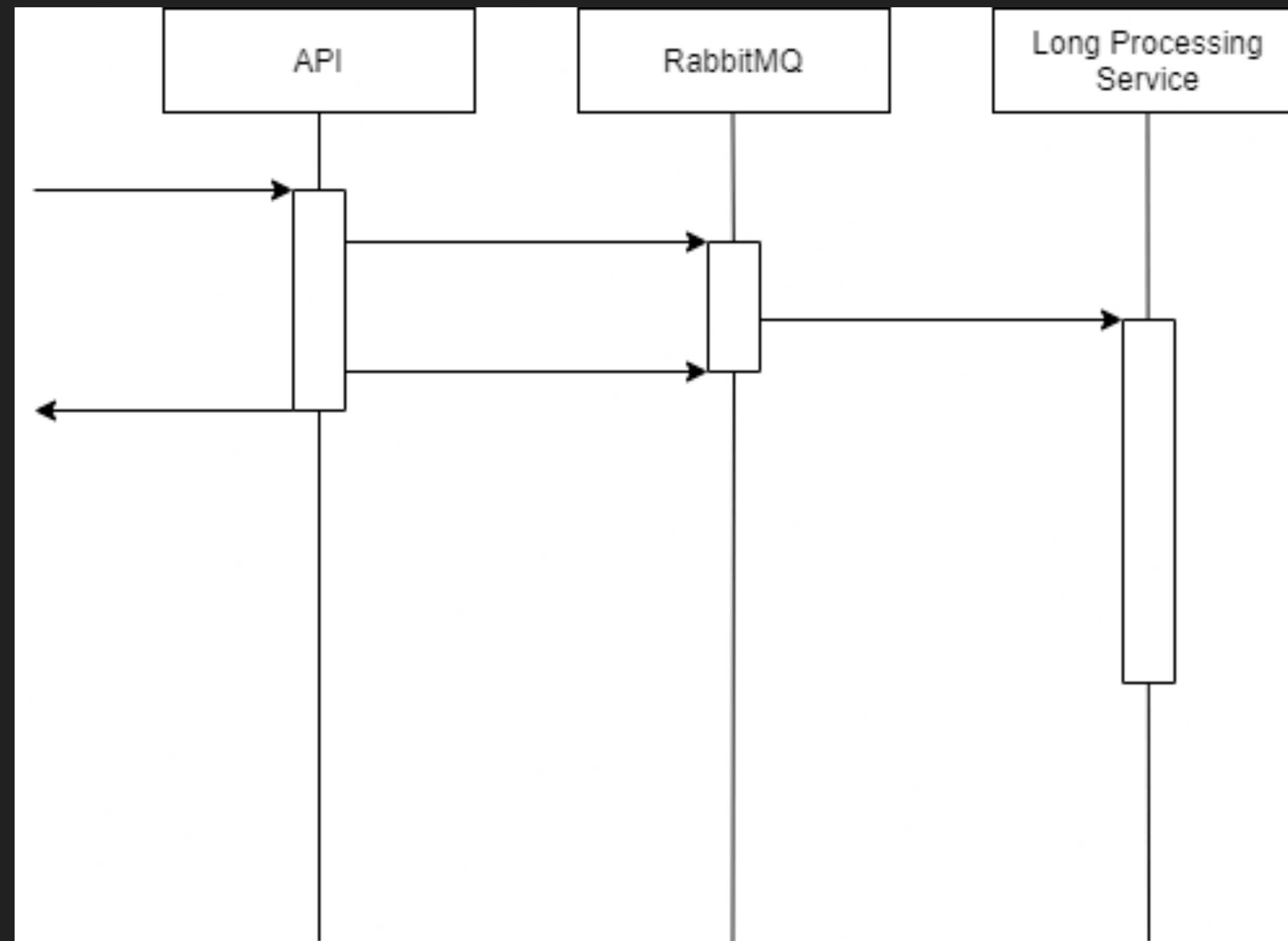
Zasada działania w dużym uproszczeniu

## ZALETY SYSTEMÓW KOLEJKOWANIA

- ▶ Nadawca nie musi znać wszystkich odbiorców wiadomości - interesuje go tylko adres pośrednika wiadomości
- ▶ Możemy mieć wielu odbiorców dla tej samej wiadomości
- ▶ Możemy w łatwy sposób dodać nowych odbiorców bez żadnych zmian w aplikacji wysyłającej wiadomości
- ▶ Wiadomości mogą czekać w kolejce w przypadku kiedy aktualnie nie ma żadnego odbiorcy, który może je przetworzyć

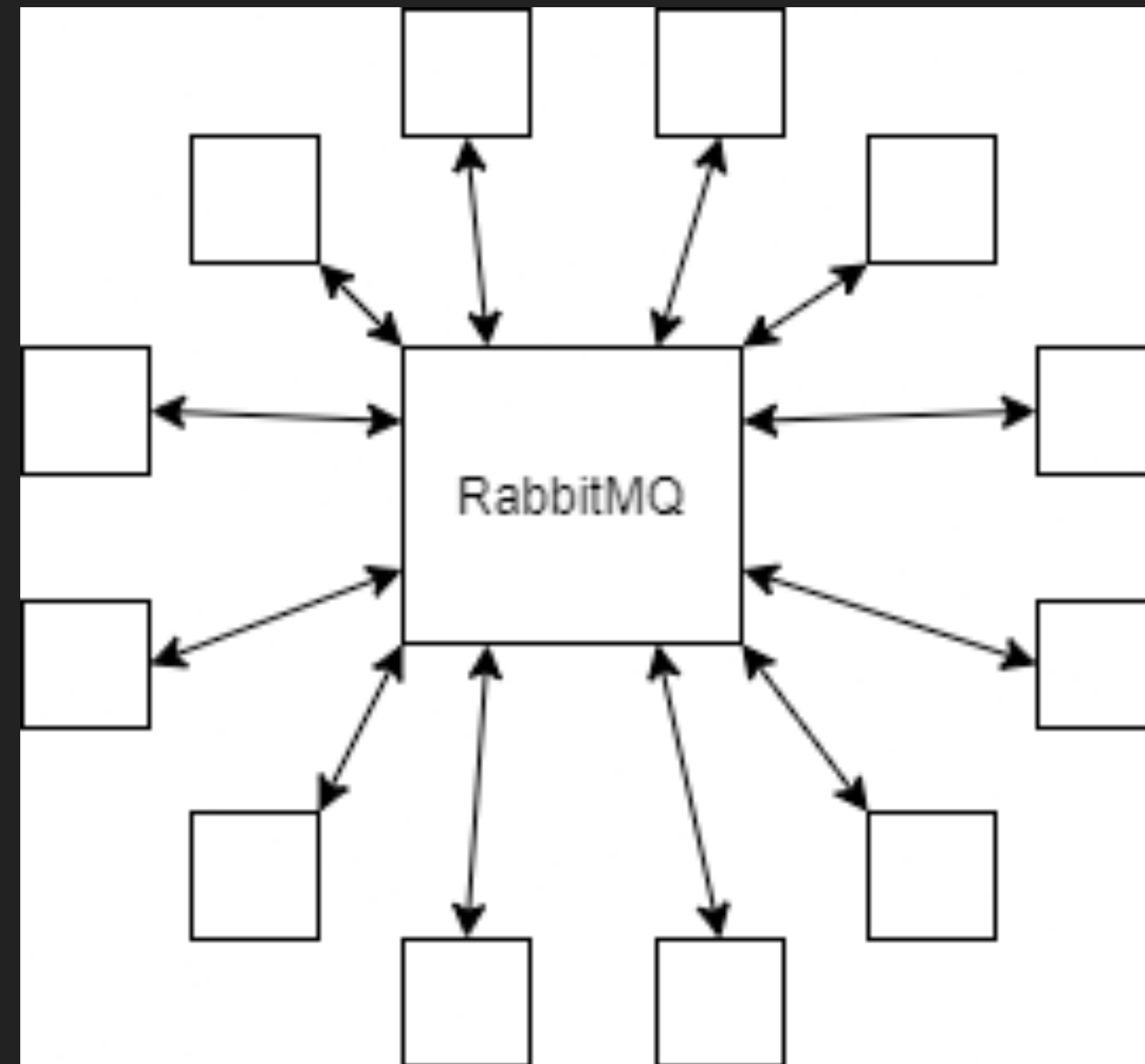
## PRZYPADKI UŻYCIA

### ► API



## PRZYPADKI UŻYCIA

### ► Mikroserwisy

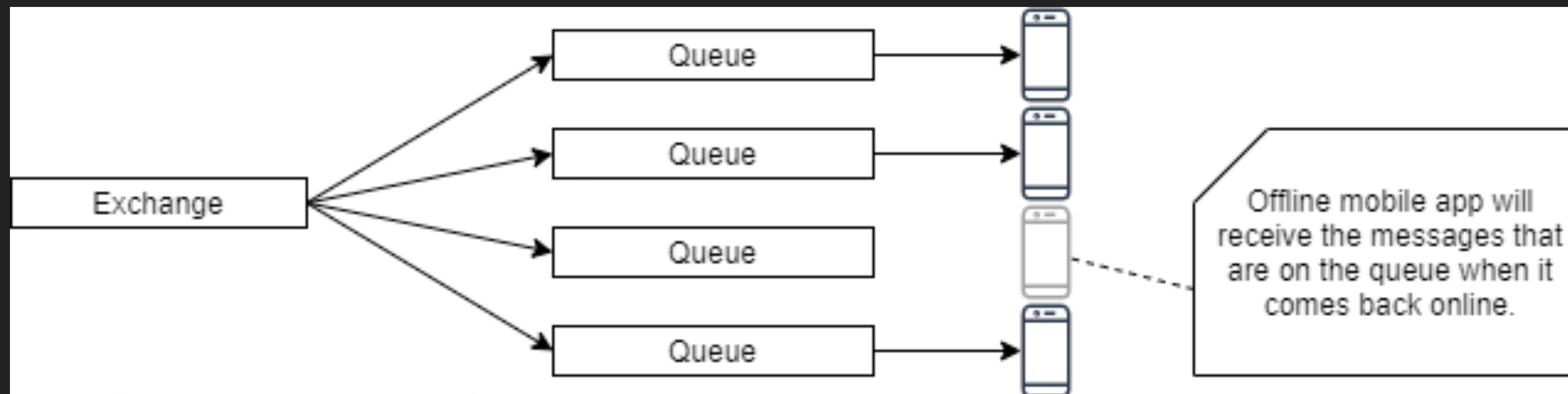


<https://www.petormlion.com/wp-content/uploads/2020/05/RabbitMQ-microservices.png>



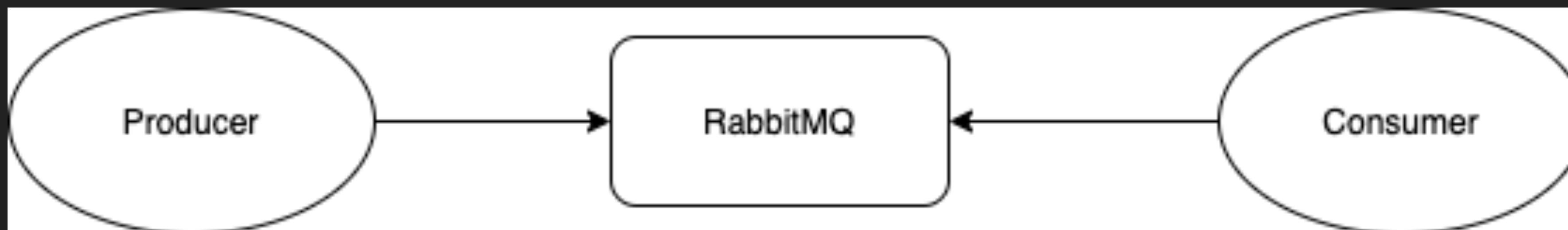
## PRZYPADKI UŻYCIA

### ► Aplikacje mobilne



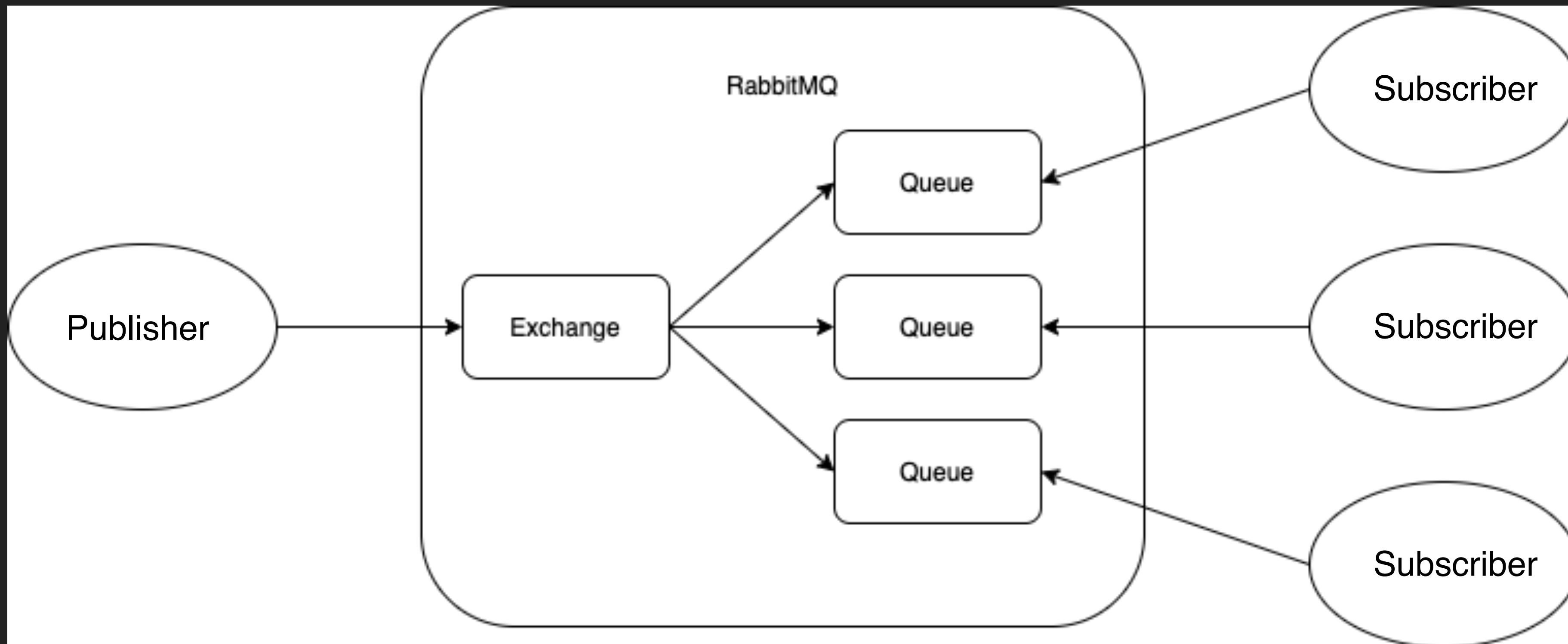
<https://www.petormorlion.com/wp-content/uploads/2020/05/RabbitMQ-mobile-apps.png>

# WYSYŁKA I ODBIÓR WIADOMOŚCI



Zasada działania w dużym uproszczeniu

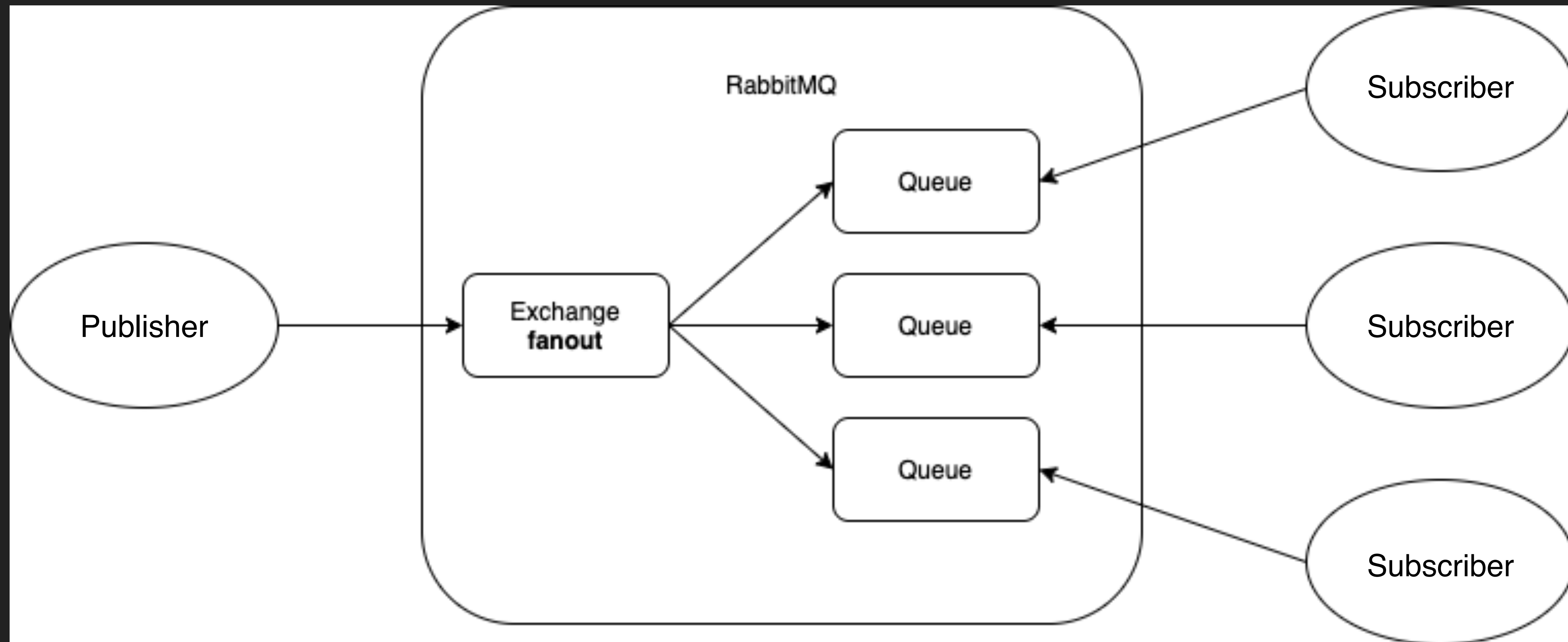
# CENTRALE WIADOMOŚCI



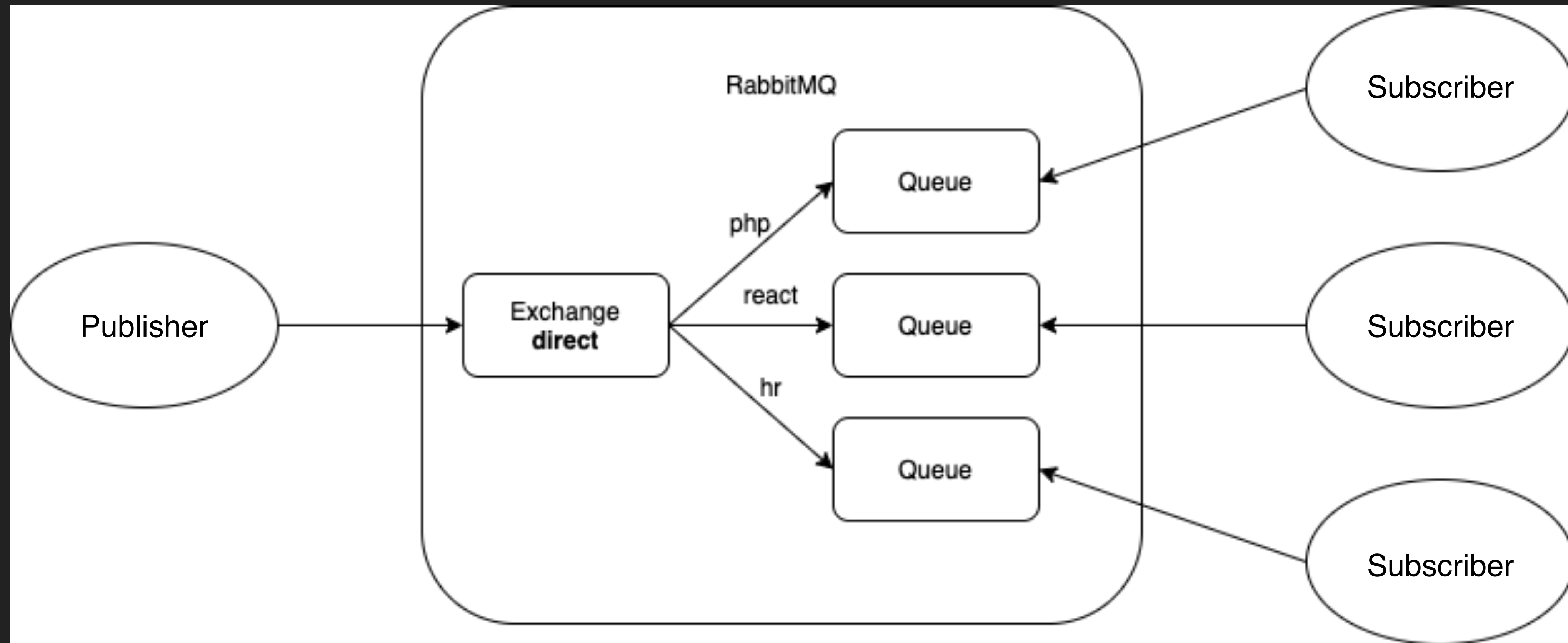
## TYPY CENTRALI WIADOMOŚCI

- ▶ fanout
- ▶ direct
- ▶ topic
- ▶ headers

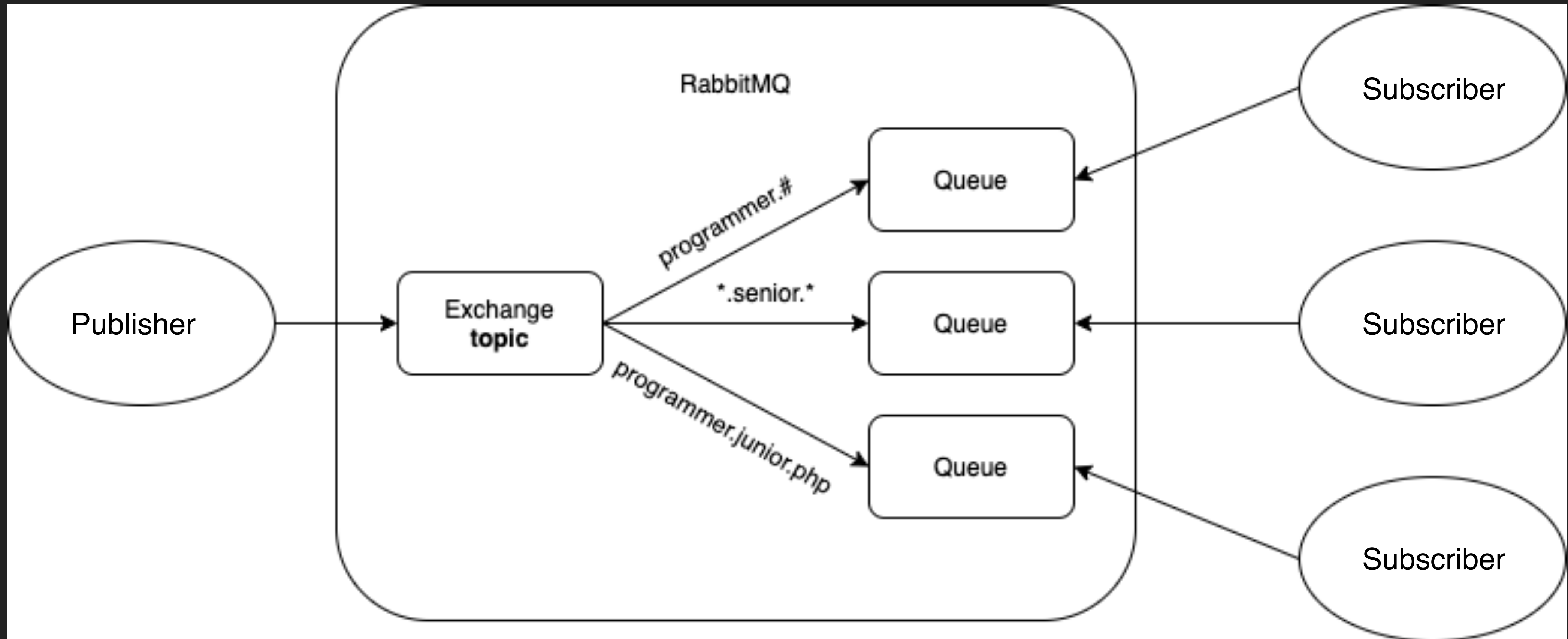
# CENTRALA TYPU FANOUT



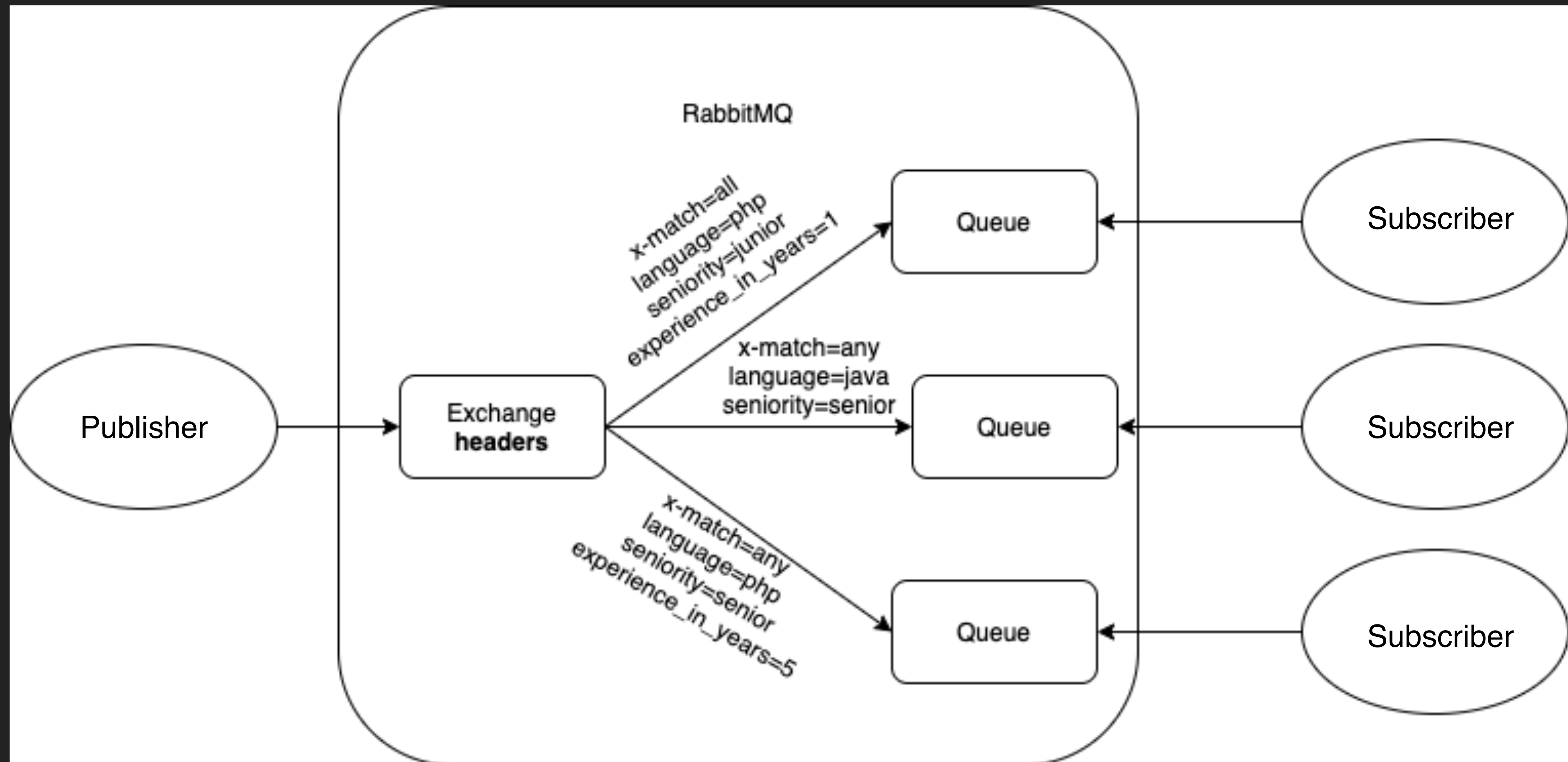
# CENTRALA TYPU DIRECT



# CENTRALA TYPU TOPIC



# CENTRAL TYPED HEADERS

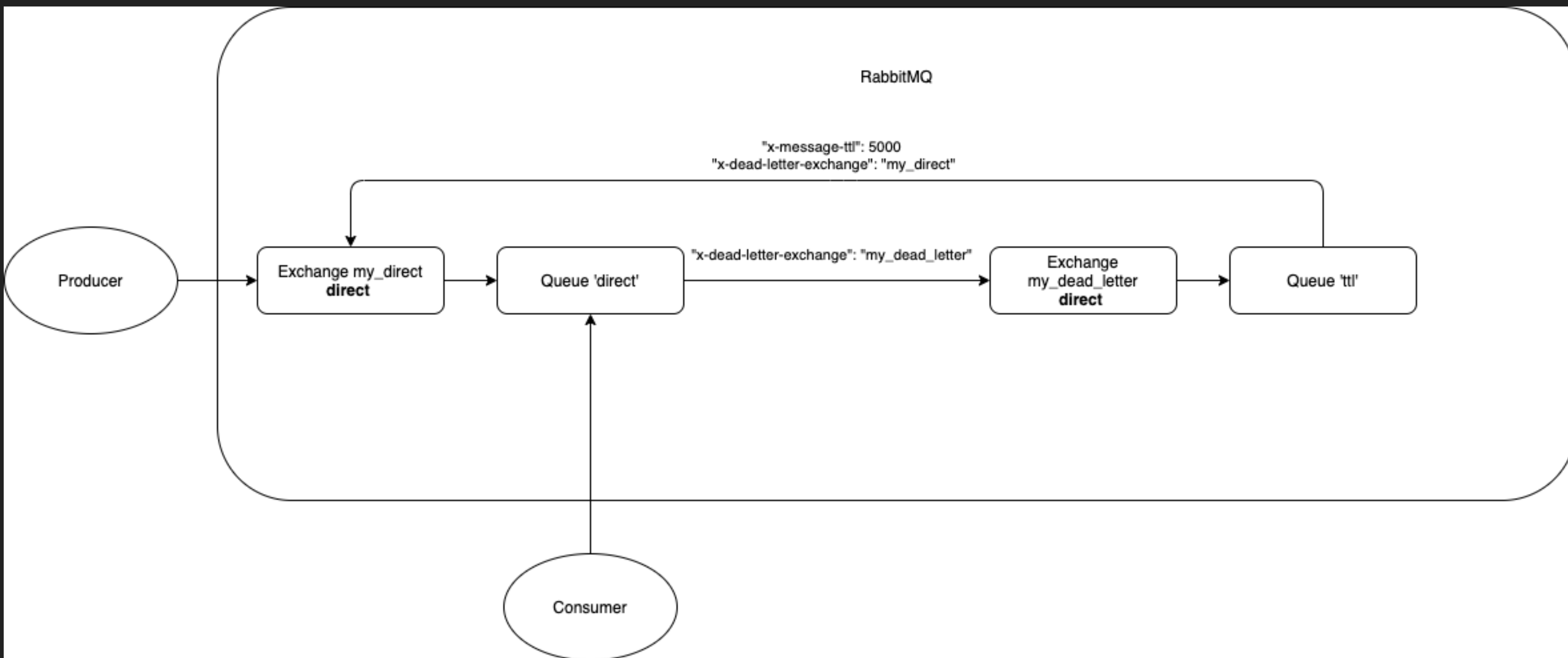




## OPÓŹNIANIE POWROTU WIADOMOŚCI DO KOLEJKI – DEAD LETTER EXCHANGES

- ▶ Przypadki kiedy wiadomość może zostać ponownie wysłana do centrali wiadomości:
  - ▶ Wiadomość potwierdzona ale z odpowiedzią negatywną (nack, reject)
  - ▶ Wiadomość wygasła (x-message-ttl)
  - ▶ Kolejka ma narzucony limit wiadomości i nie przyjmuje kolejnych wiadomości (x-max-length, x-max-length-bytes)

# OPÓŹNIANIE POWROTU WIADOMOŚCI DO KOLEJKI – DEAD LETTER EXCHANGES



## GOTOWE PAKIETY DO WYKORZYSTANIA Z SYMFONY I LARAVELEM

- ▶ Symfony Messenger
- ▶ Swarrot (Symfony)
- ▶ rabbit-mq-admin-toolkit (Symfony)
- ▶ laravel-queue-rabbitmq (Laravel)

# SYMFONY MESSENGER

```
1  # .env
2  MESSENGER_TRANSPORT_DSN=amqp://guest:guest@localhost:5672/%2f/messages
3
```

```
framework:
  messenger:
    transports:
      amqp_email_notification:
        dsn: '%env(MESSENGER_TRANSPORT_DSN)%'
        options:
          exchange:
            name: email_notifications
            type: fanout
          queues:
            email_notifications: ~
          retry_strategy:
            max_retries: 3
            delay: 1000
            multiplier: 2
            max_delay: 0
    routing:
      App\Message\EmailNotification: amqp_email_notification
```

# RABBIT-MQ-ADMIN-TOOLKIT

```
/:  
  exchanges:  
    my_direct:  
      type: direct  
      durable: true  
    my_dead_letter:  
      type: direct  
      durable: true  
  queues:  
    direct:  
      durable: true  
      bindings:  
        - exchange: my_direct  
      arguments:  
        x-dead-letter-exchange: my_dead_letter  
  ttl:  
    durable: true  
    bindings:  
      - exchange: my_dead_letter  
    arguments:  
      x-message-ttl: 5000  
      x-dead-letter-exchange: my_direct
```

# SWARROT

```
swarrot:
  provider: pecl
  default_connection: rabbitmq
  logger: logger
  connections:
    rabbitmq:
      url: '%env(resolve:RABBITMQ_URL)%'
  consumers:
    whitewalker_lucie_consumer:
      processor: App\Consumer\ThirdPartyConsumer
      middleware_stack:
        - configurator: swarrot.processor.signal_handler
        - configurator: swarrot.processor.max_messages
          extras:
            max_messages: '%env(int:CONSUMER_MAX_MESSAGES)%'
        - configurator: swarrot.processor.max_execution_time
          extras:
            max_execution_time: '%env(int:CONSUMER_MAX_EXECUTION_TIME)%'
        - configurator: swarrot.processor.ack
        - configurator: swarrot.processor.retry
          extras:
            retry_exchange: whitewalker_lucie_retry
            retry_attempts: 5
            retry_routing_key_pattern: 'whitewalker_lucie_retry_%%attempt%%'
```

# LARAVEL-QUEUE-RABBITMQ

```
'connections' => [  
    // ...  
  
    'rabbitmq' => [  
        // ...  
  
        'options' => [  
            'queue' => [  
                // ...  
  
                'exchange' => 'application-x',  
                'exchange_type' => 'topic',  
                'exchange_routing_key' => '',  
            ],  
        ],  
    ],  
    // ...  
],
```

```
'connections' => [  
    // ...  
  
    'rabbitmq' => [  
        // ...  
  
        'options' => [  
            'queue' => [  
                // ...  
  
                'job' => \App\Queue\Jobs\RabbitMQJob::class,  
            ],  
        ],  
    ],  
    // ...  
],
```



**DZIĘKUJĘ ZA UWAGĘ**



- ▶ <https://www.rabbitmq.com>
- ▶ <https://www.petermorlion.com/rabbitmq-use-cases/>
- ▶ <https://packagist.org/>
- ▶ <https://symfony.com/doc/current/messenger.html>