

Hello!

Anton Davydov

github: [davydovanton](#)
twitter: [anton_davydov](#)

 **RCNTEC developer**

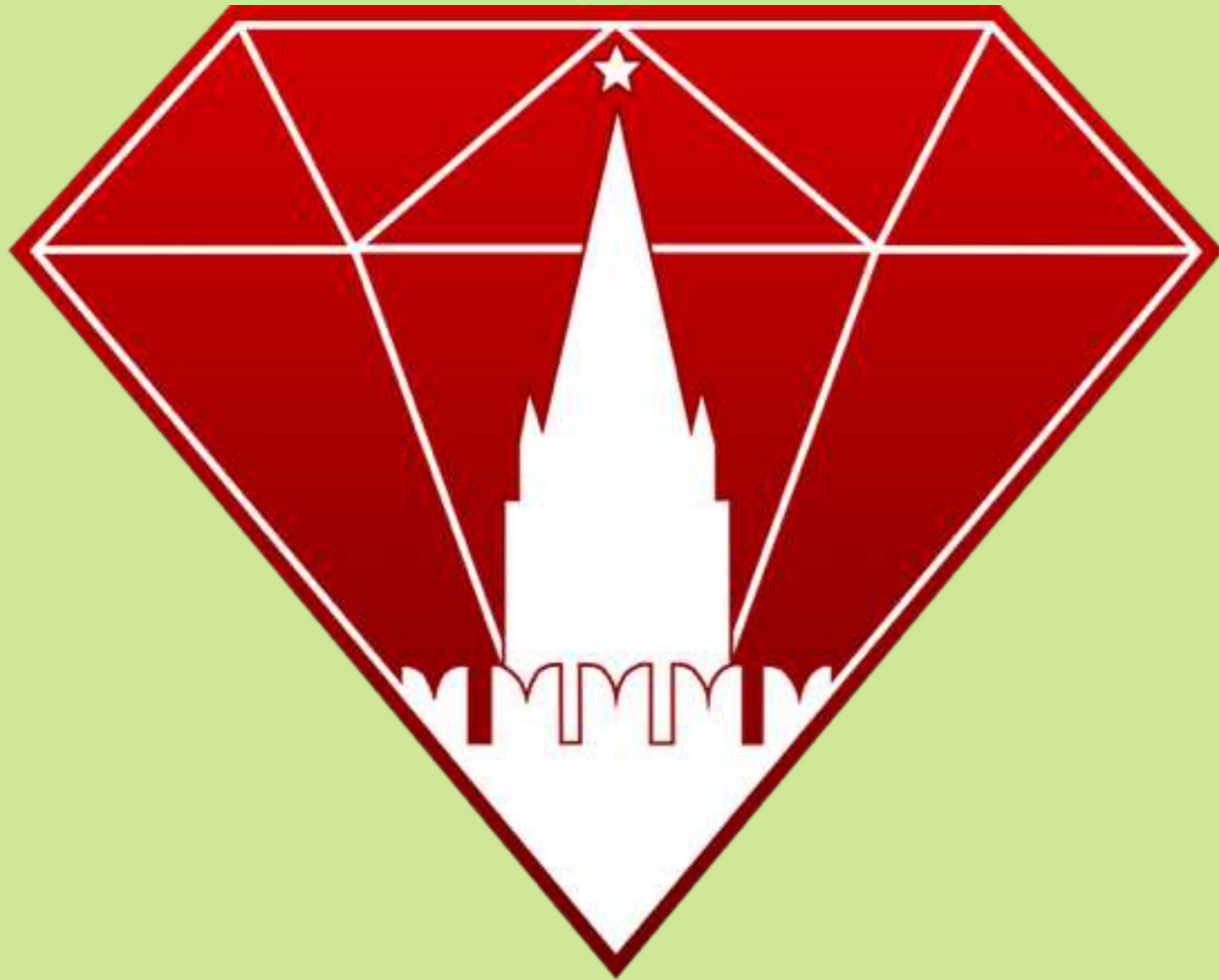
 **sidekiq commiter**

 **ruby doc / rails / crystal**

 **OSS one love <3**

@rubyunderhood curator

<https://twitter.com/rubyunderhood>



@moscowrb

<http://moscow-rb.org>

You



Beer

You

Me

Beer



I (we) < 3 sidekiq!



Simple, efficient background processing for Ruby.

Sidekiq server

New thread

Signals:

INT

TERM

USR1

USR2

TTIN

I (we) < 3 actors!

Manager actor

Manager actor



Fetcher actors

Manager actor

Fetcher actors

Redis

Manager actor

```
graph TD; Manager[Manager actor] --> Fetcher[Fetcher actors]; Manager --> Processor[Processor actors]; Fetcher --> Redis((Redis));
```

Fetcher actors

Processor actors

Redis

Manager actor

Fetcher actors

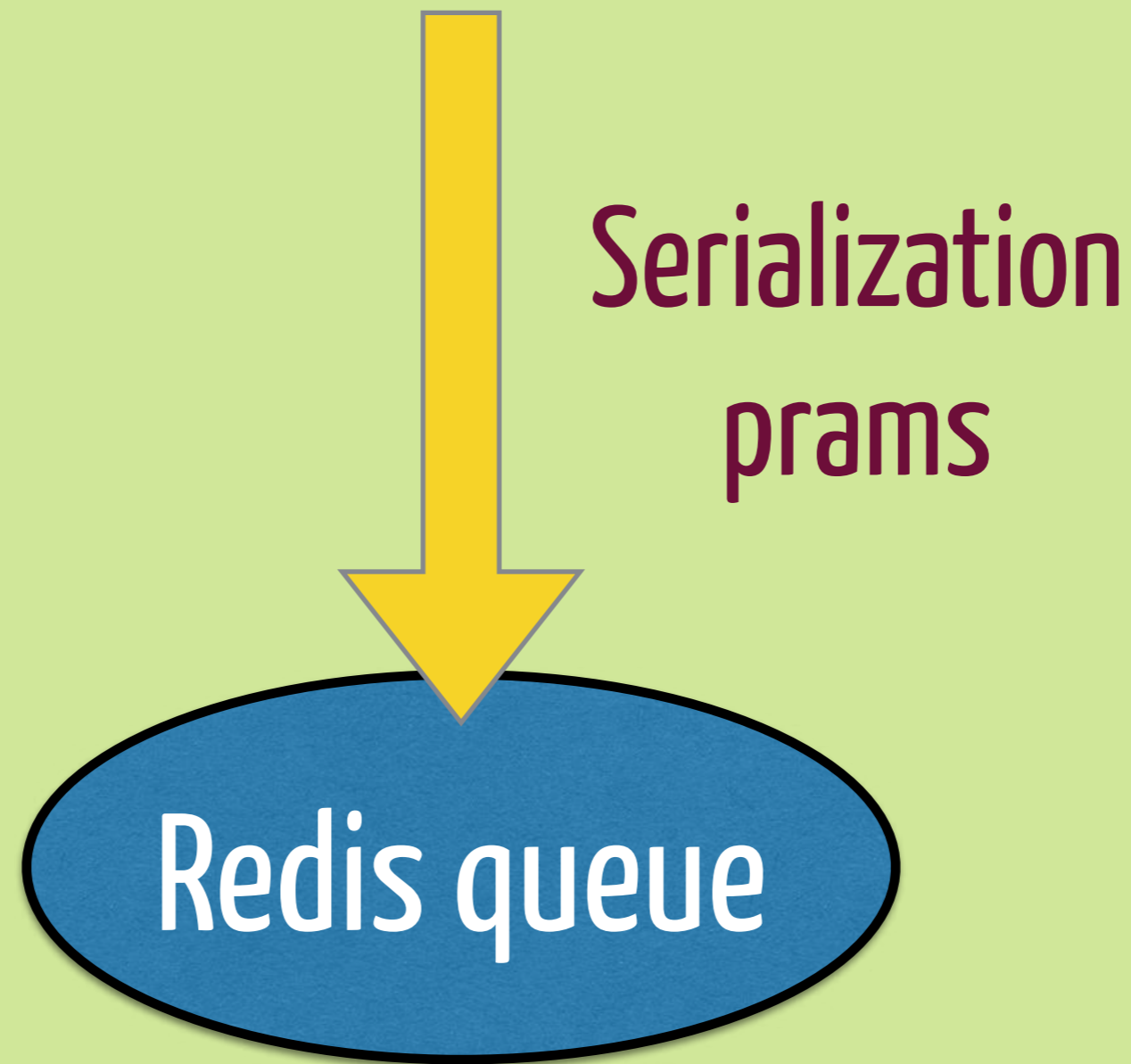
Redis

Processor actors

Workers

`MyWorker.perform_async`

```
irb > MyWorker.perform_async
```



Manager actor

Fetcher actors

Processor actors

Redis

Workers

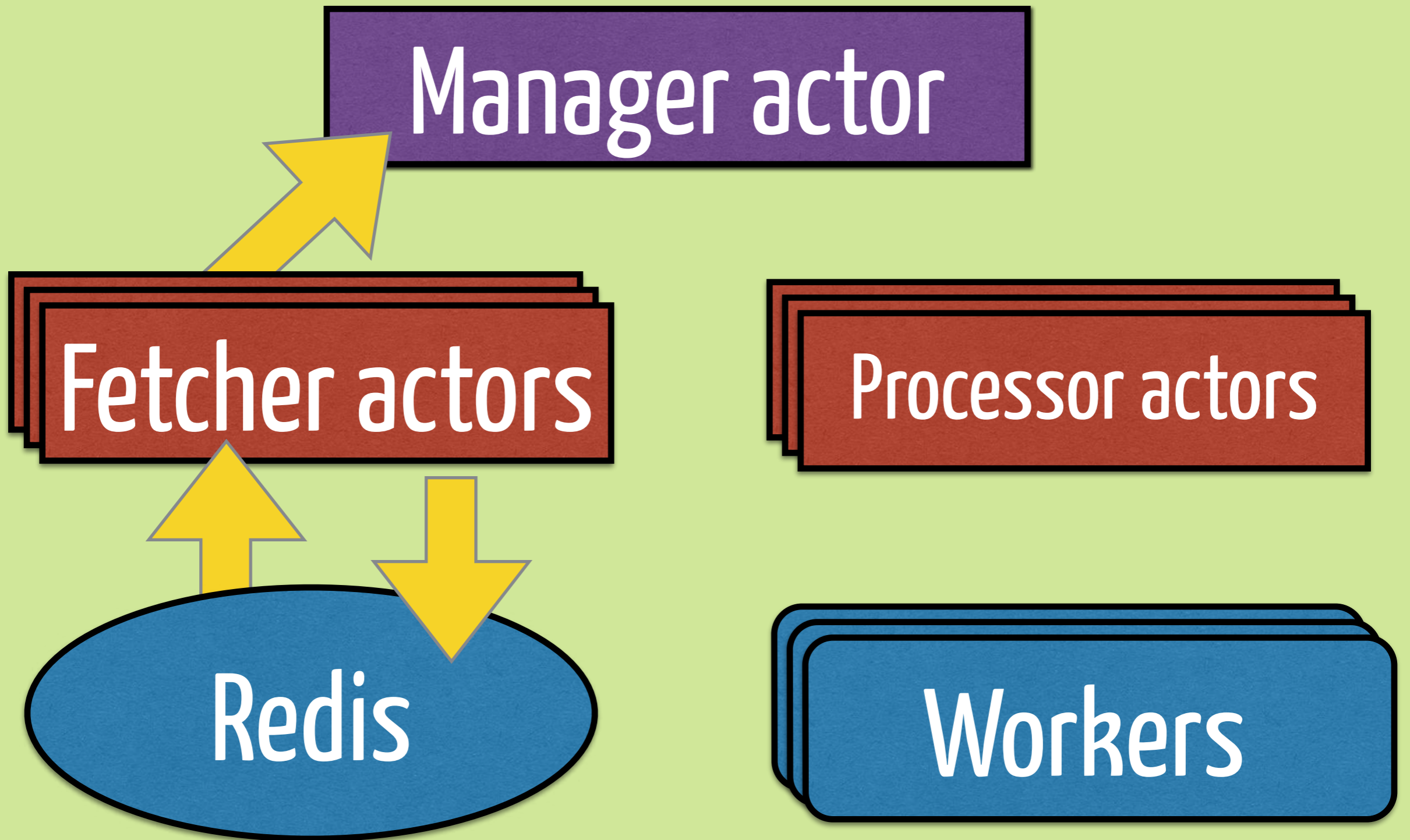
Manager actor

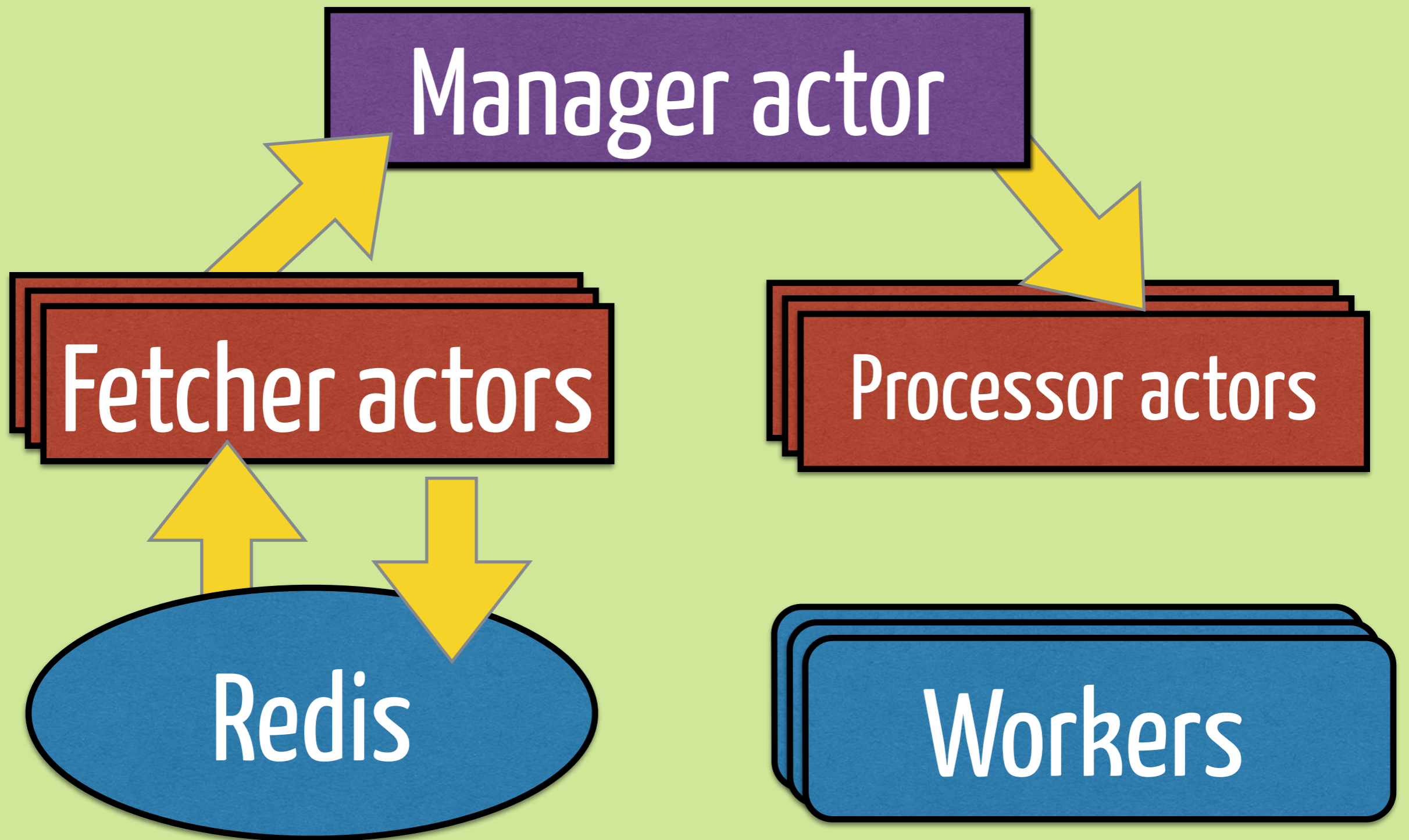
Fetcher actors

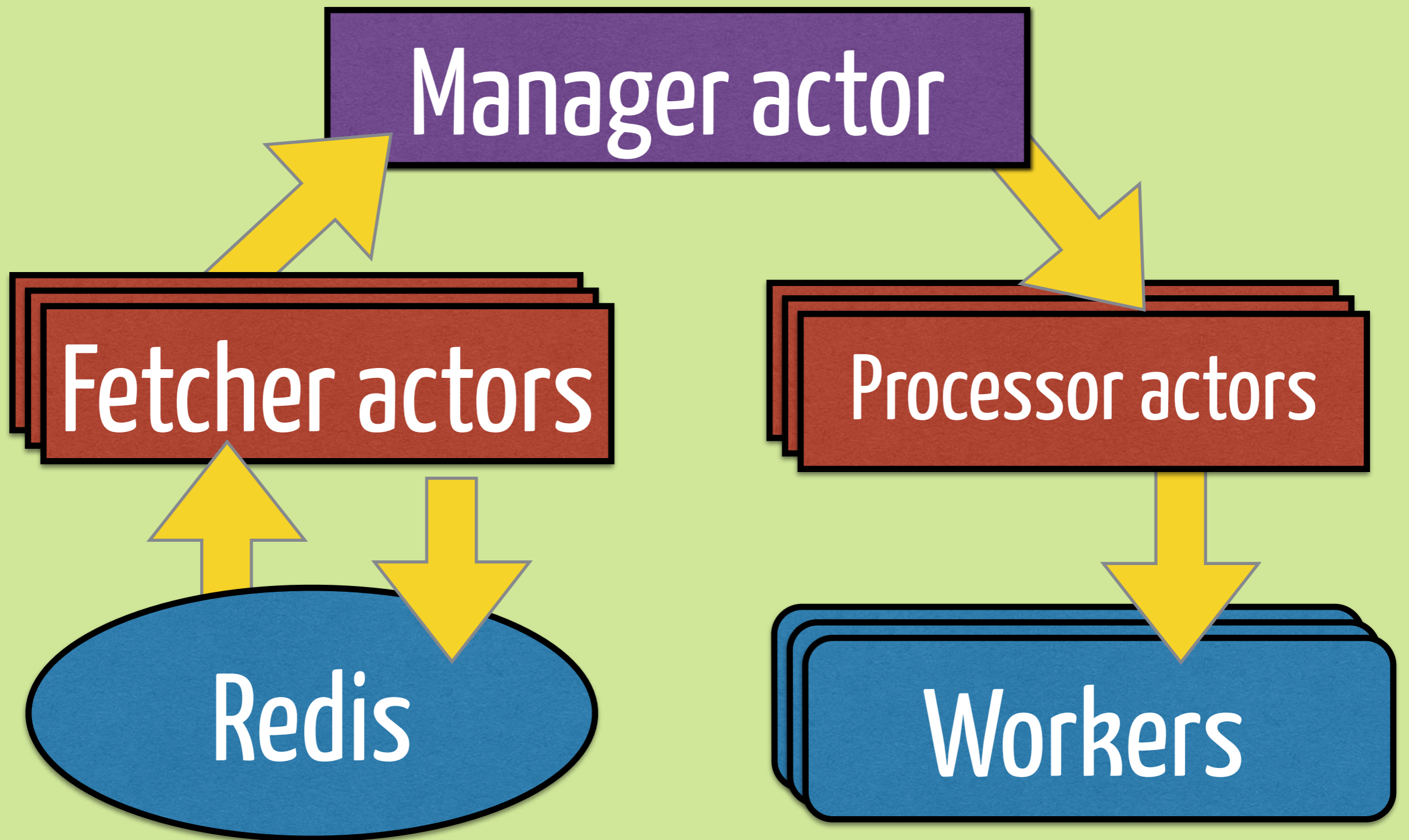
Processor actors

Redis

Workers







LongWorker

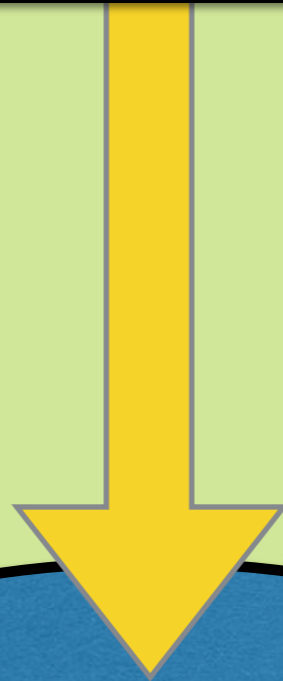
Redis

After 8 seconds

LongWorker

Serialization
prams

Redis



middleware

```
Sidekiq.configure_server do |config|
  config.server_middleware do |chain|
    chain.add YourMiddleware, options
  end
end
```

```
def call(worker, msg, queue)
  # code before call you worker
  yield
rescue => e
  # run if your worker is failed
  raise e
ensure
  # run in your worker passed or failed
end
```

```
def call(worker, msg, queue)
  # code before call you worker
  yield
rescue => e
  # run if your worker is failed
  raise e
ensure
  # run in your worker passed or failed
end
```

```
def call(worker, msg, queue)
  # code before call you worker
  yield
rescue => e
  # run if your worker is failed
  raise e
ensure
  # run in your worker passed or failed
end
```

```
def call(worker, msg, queue)
  # code before call you worker
  yield
rescue => e
  # run if your worker is failed
  raise e
ensure
  # run in your worker passed or failed
end
```

Sidekiq-statistic

github.com/davydovanton/sidekiq-statistic

Improved display of statistics for
your sidekiq workers and jobs.

Middleware



Calculate



Web UI

JSON

New Relic

Middleware



redis

Web UI

JSON

New Relic



Calculate





Worker name



Last job status



Runtime



Start time

 Jobs per day charts for each worker

 Last job status for each worker

 Passed and failed jobs count

 Average job time

 Max and min jobs time

 Total jobs time

 Last job time

1 < 3 feedback!

Thank you!

Anton Davydov

github.com/davydovanton

twitter.com/anton_davydov