Queue it!
What job queues can do for you!

Codepalousa 2014 - February 26th 2014 - Louisville, KY, USA

Search

5

Ole Michaelis
@CodeStars

Out for a road trip through Louisville #CPL14

Reply Delete Favorite ••• More

FAVORITE
1

4:09 AM - 25 Feb 2014 from Louisville Central, KY

Reply to @CodeStars

© 2014 Twitter About Help Ads info

Search

5

Ole Michaelis
@CodeStars

Getting pulled out by the police in US - check! #CPL14 /w @InEveryLanguage @thejmather

Reply Delete Favorite ••• More

RETWEETS 2

FAVORITES 3

4:24 AM - 25 Feb 2014 from Louisville Central, KY

Reply to @InEveryLanguage @thejmather

Ole Michaelis

@CodeStars // codestars.eu // slidr.io // So Coded Conference

Jimdo

Do You Queue?

you do, you just may not know

The Problem!

computers are slow

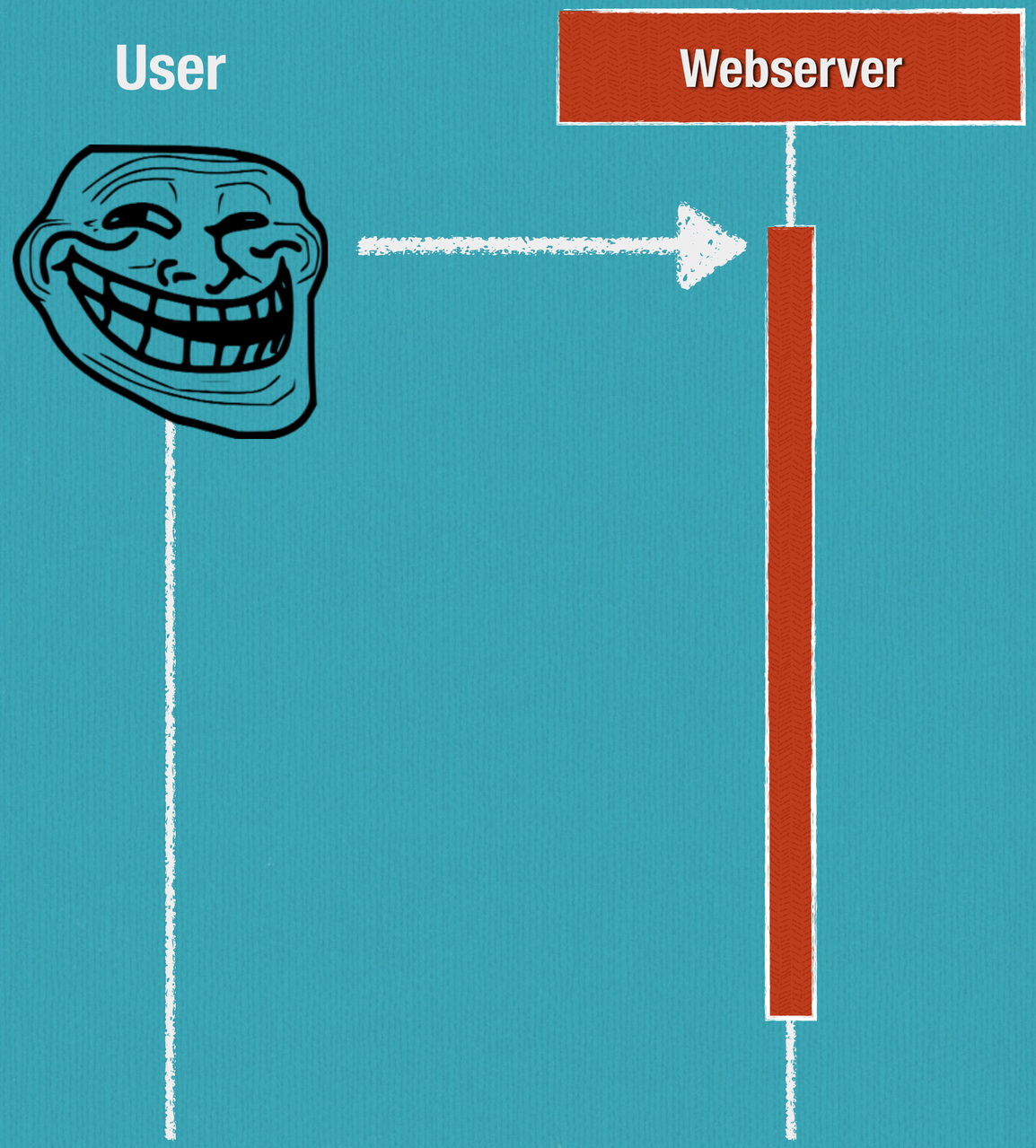
User

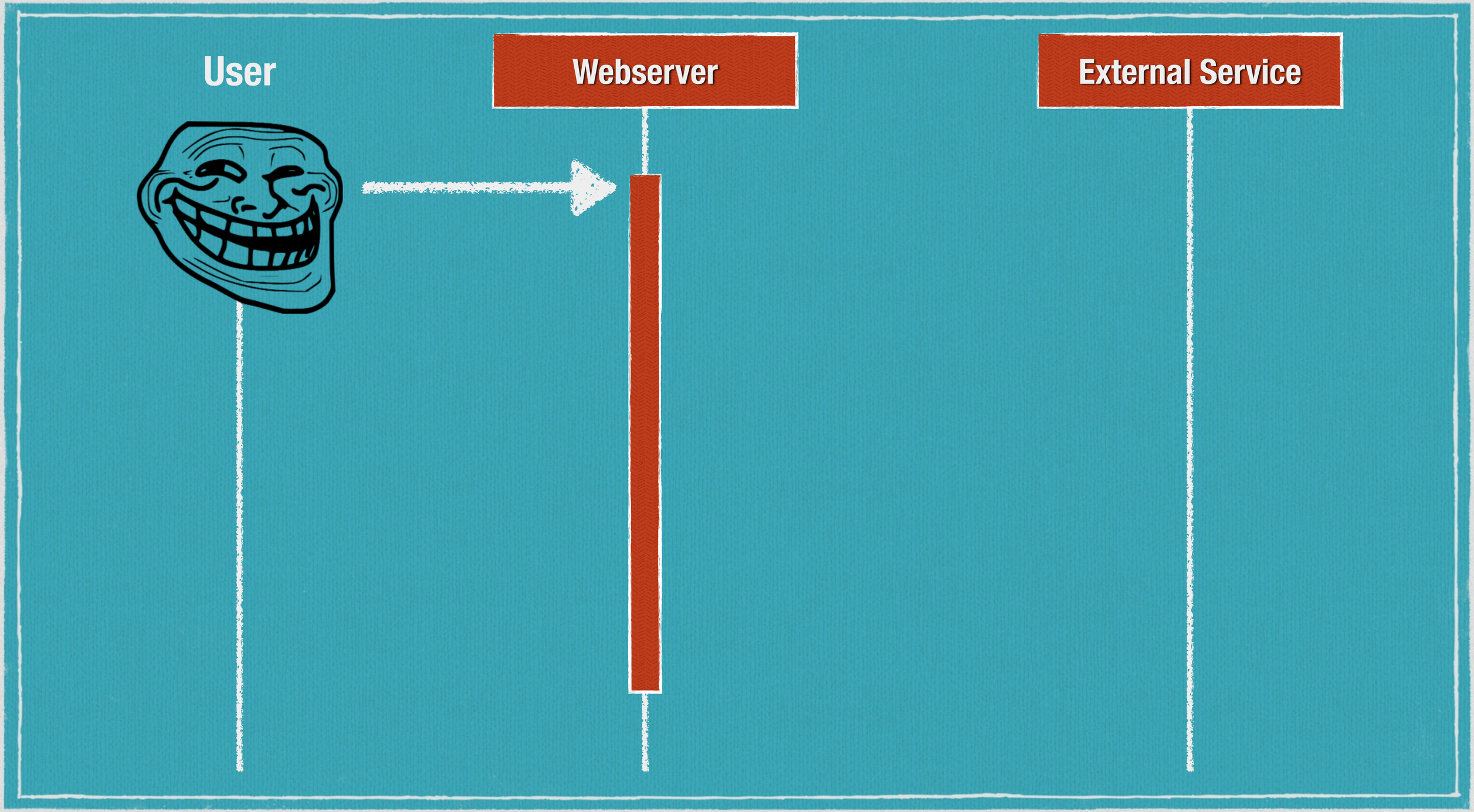
User

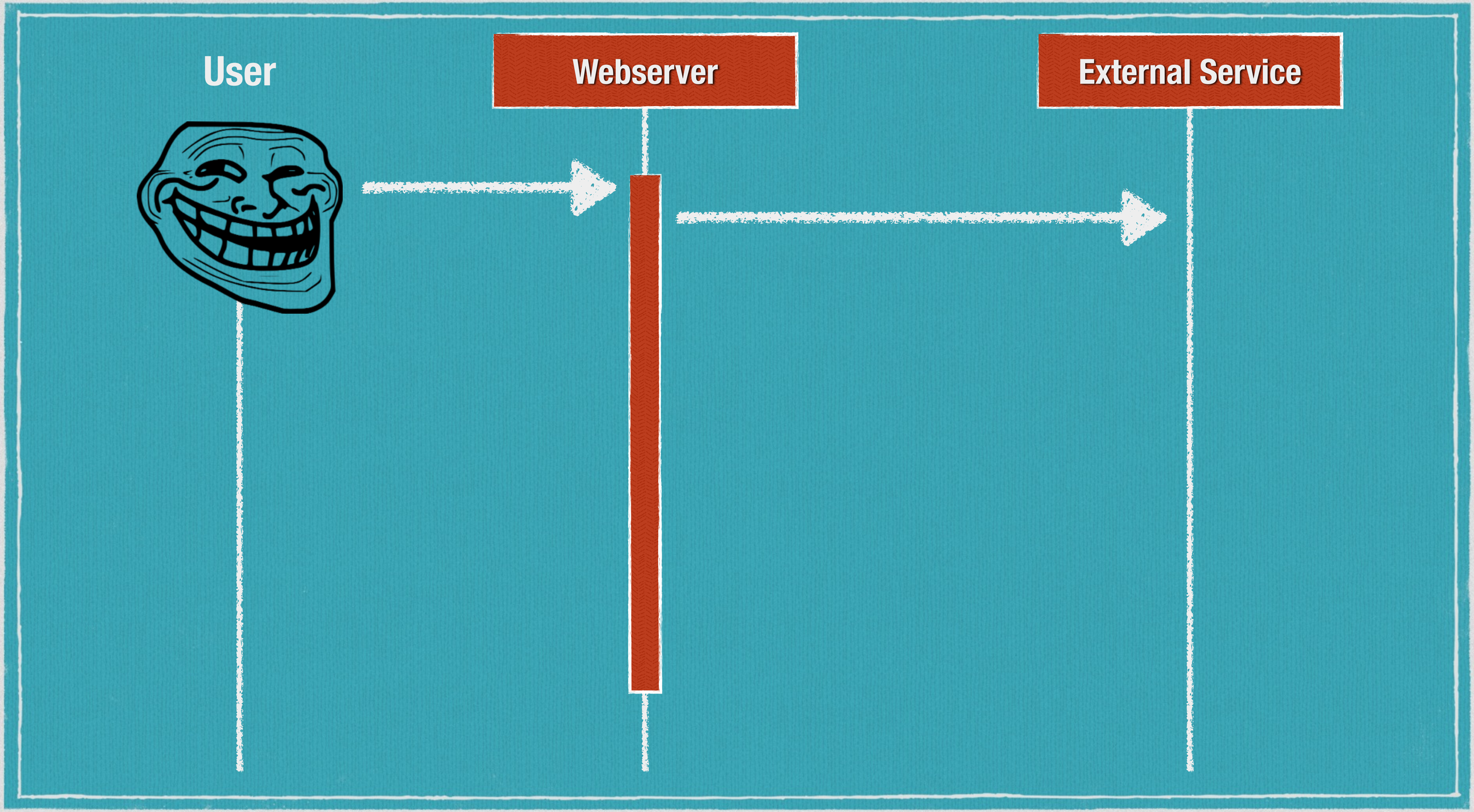
Webserver

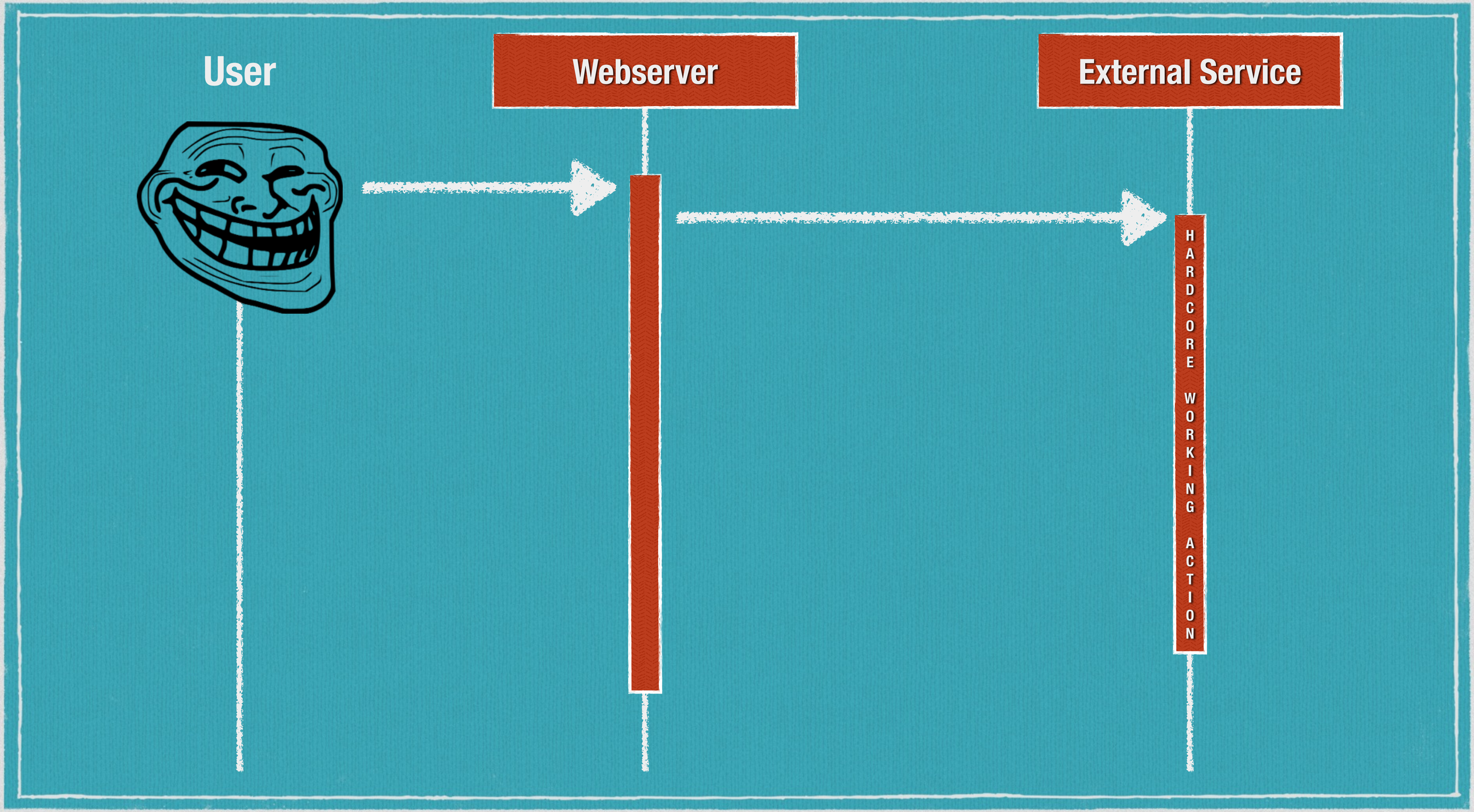
User

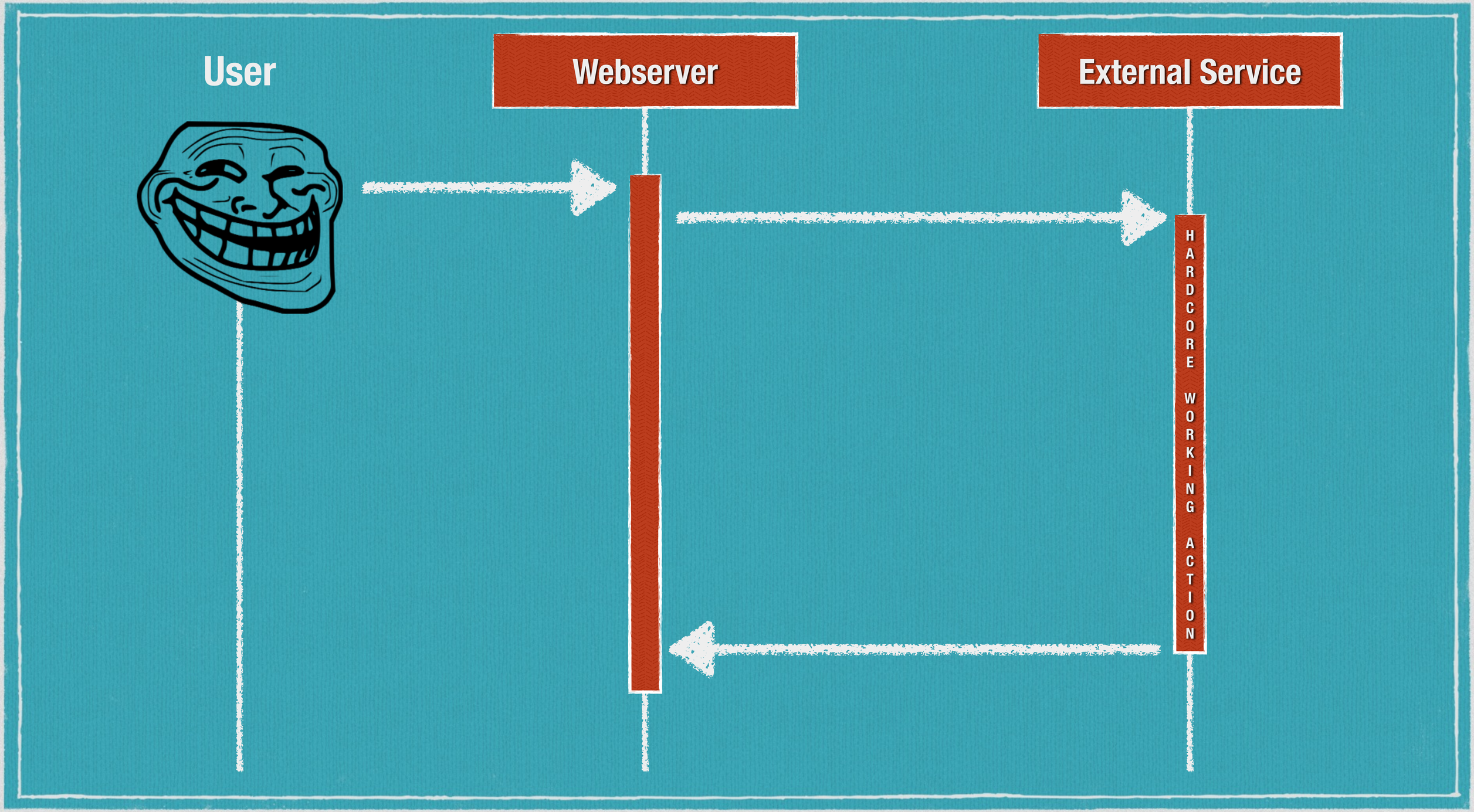
Webserver

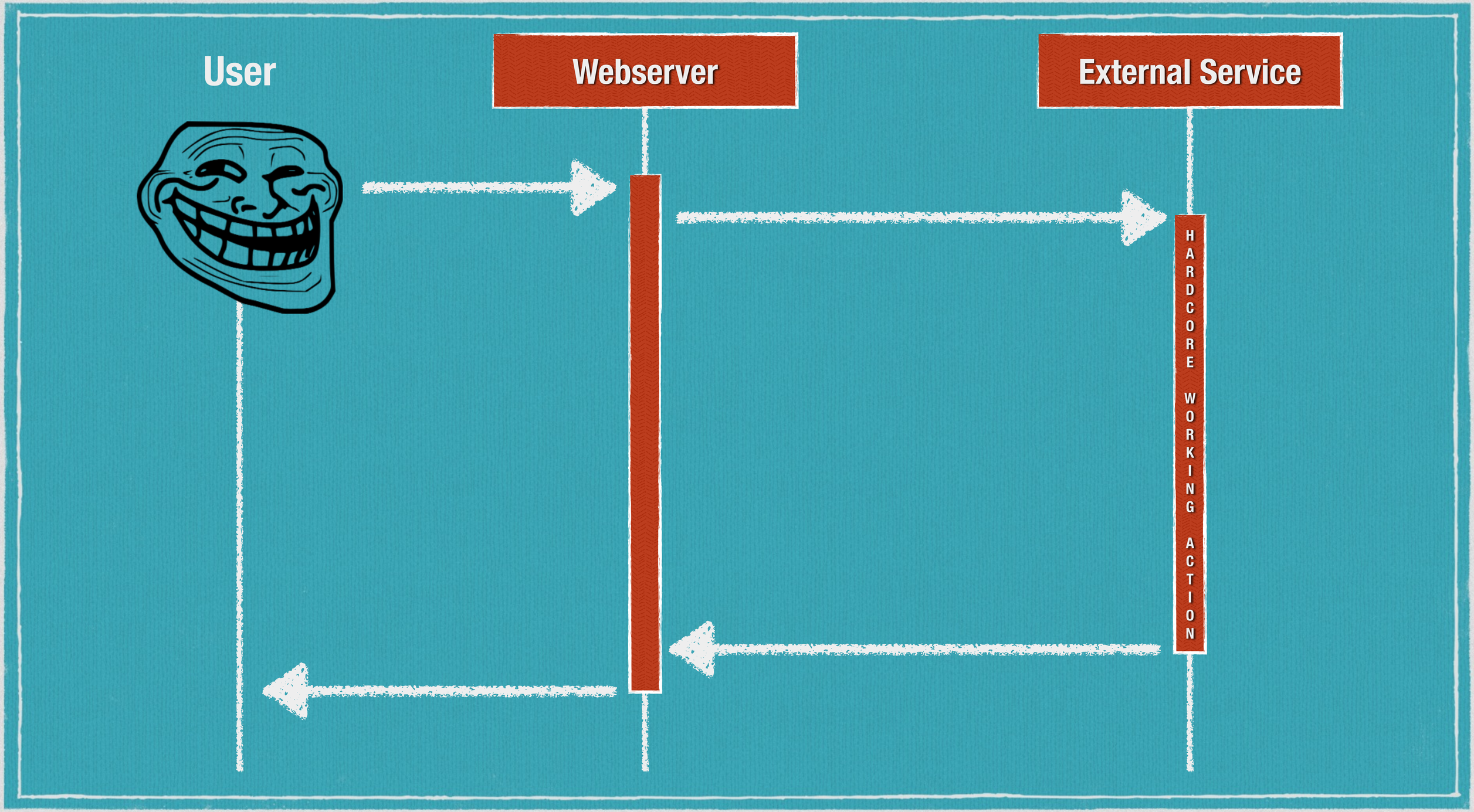

User
Webserver

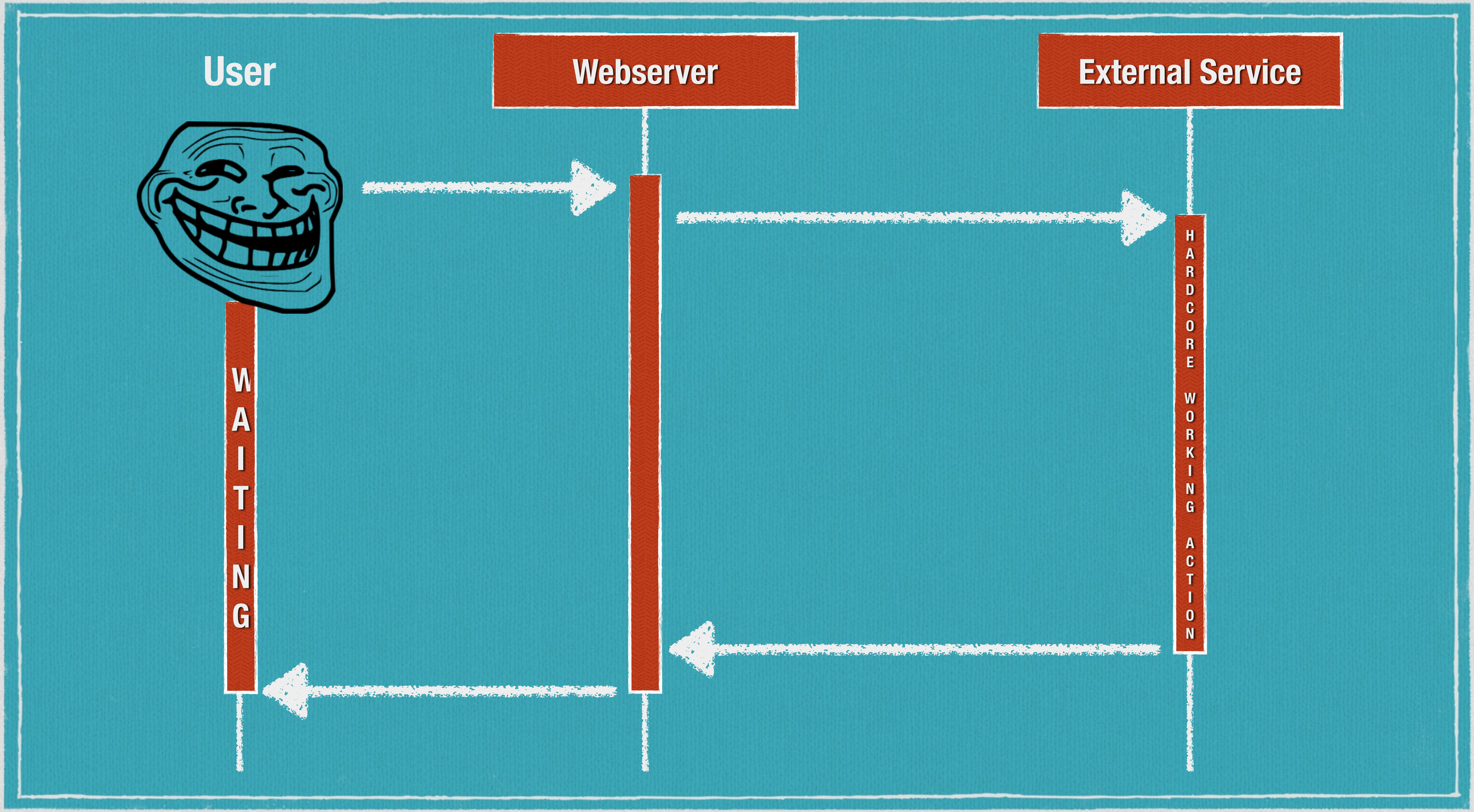

User
Webserver
External Service

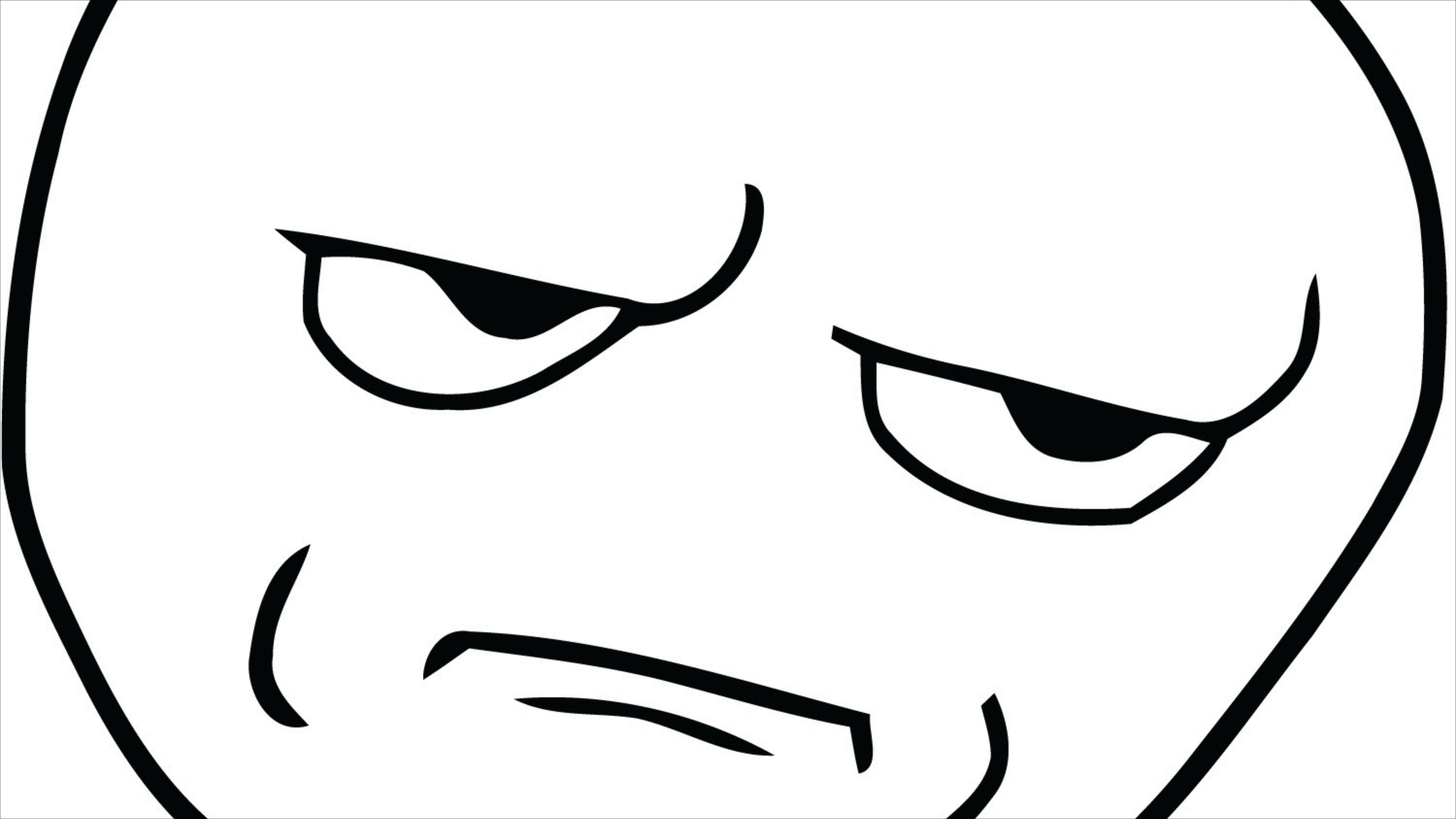

User
Webserver
External Service

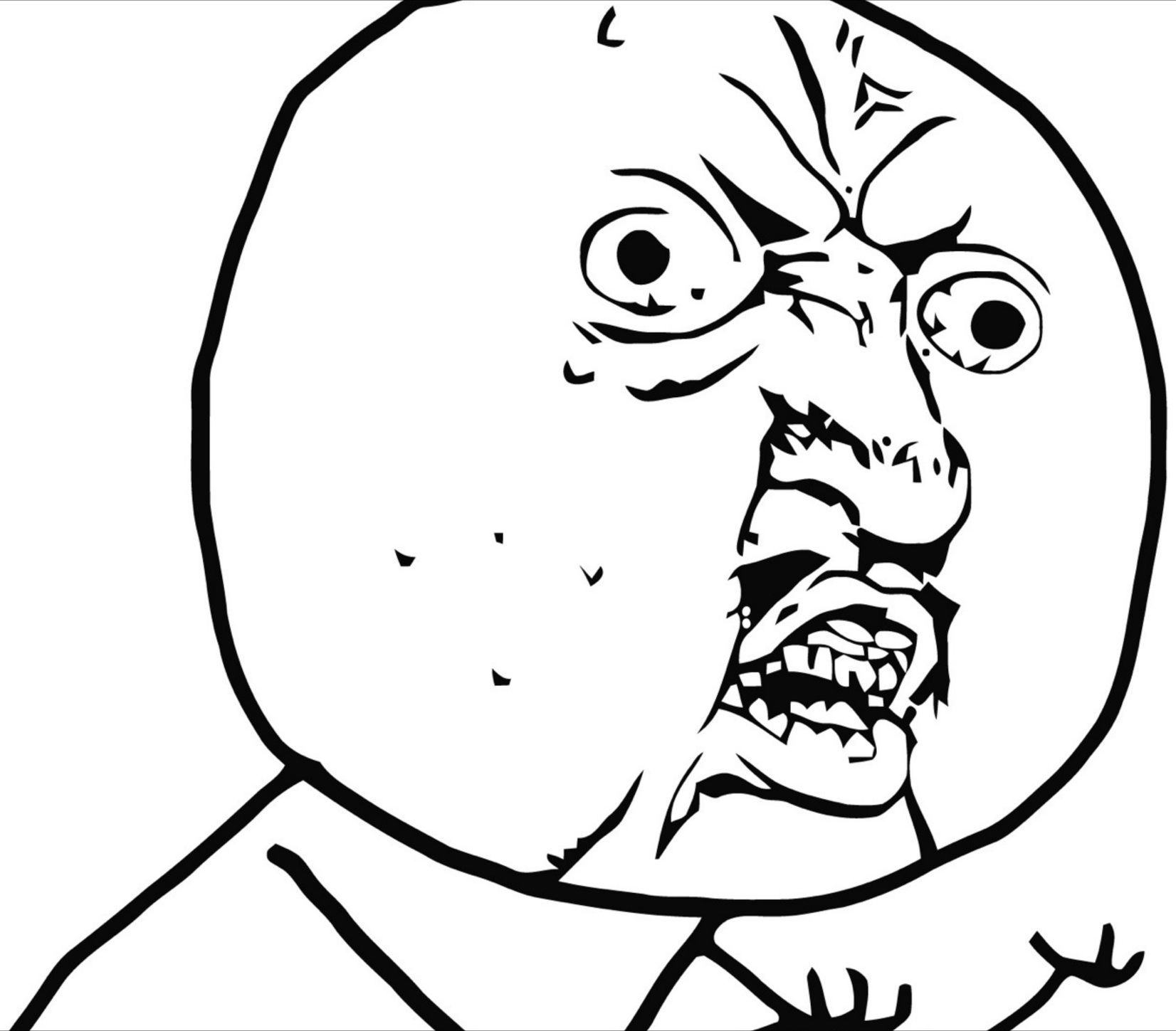

User
Webserver
External Service
HARDCORE WORKING ACTION


User
Webserver
External Service
HARDCORE WORKING ACTION


User
Webserver
External Service
HARDCORE WORKING ACTION


User
Webserver
External Service
WAITING
HARDCORE WORKING ACTION





https://secure.flickr.com/photos/vorsprach/8997923353/

Queues To The Rescue!

yay people

do it async

User

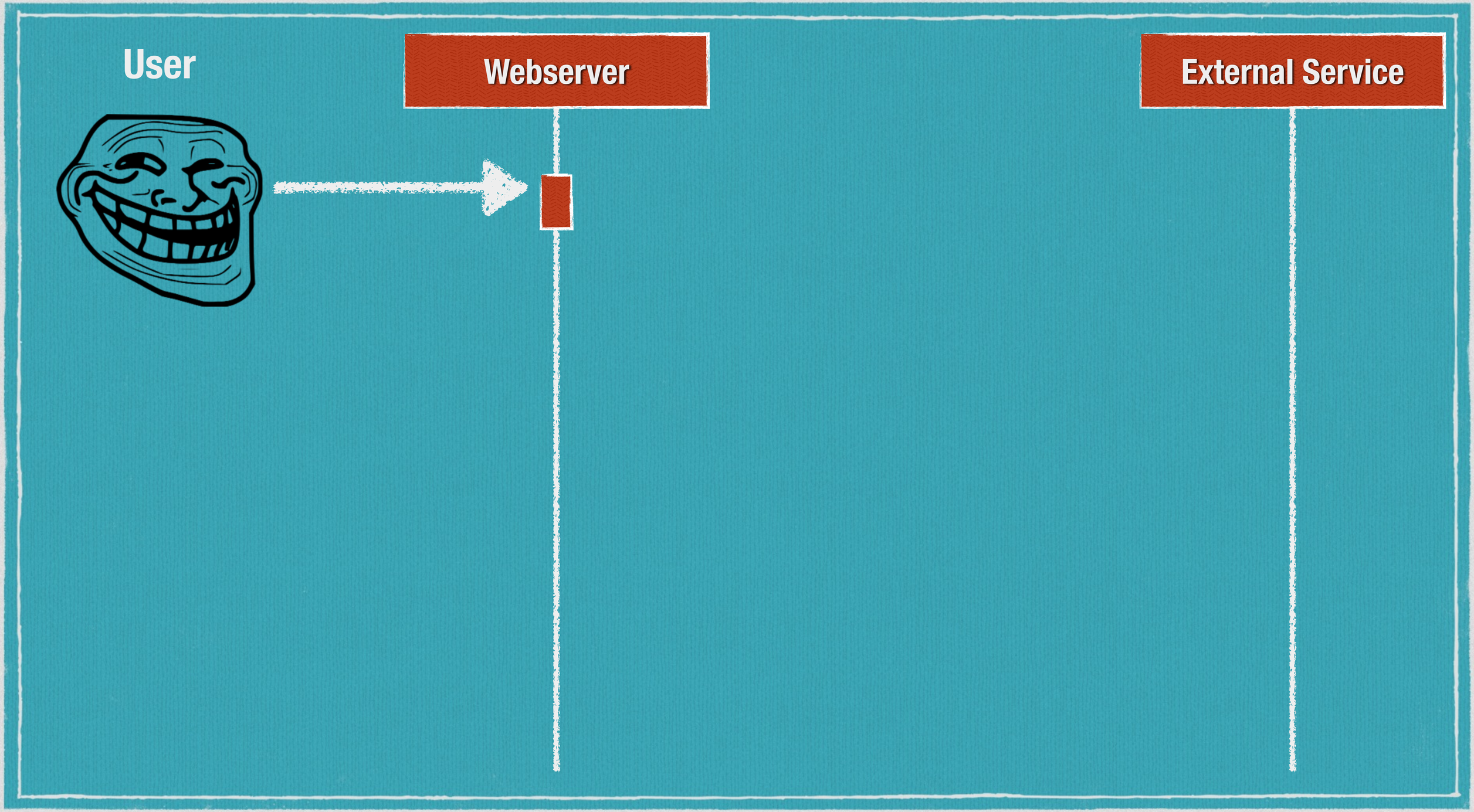
Webserver

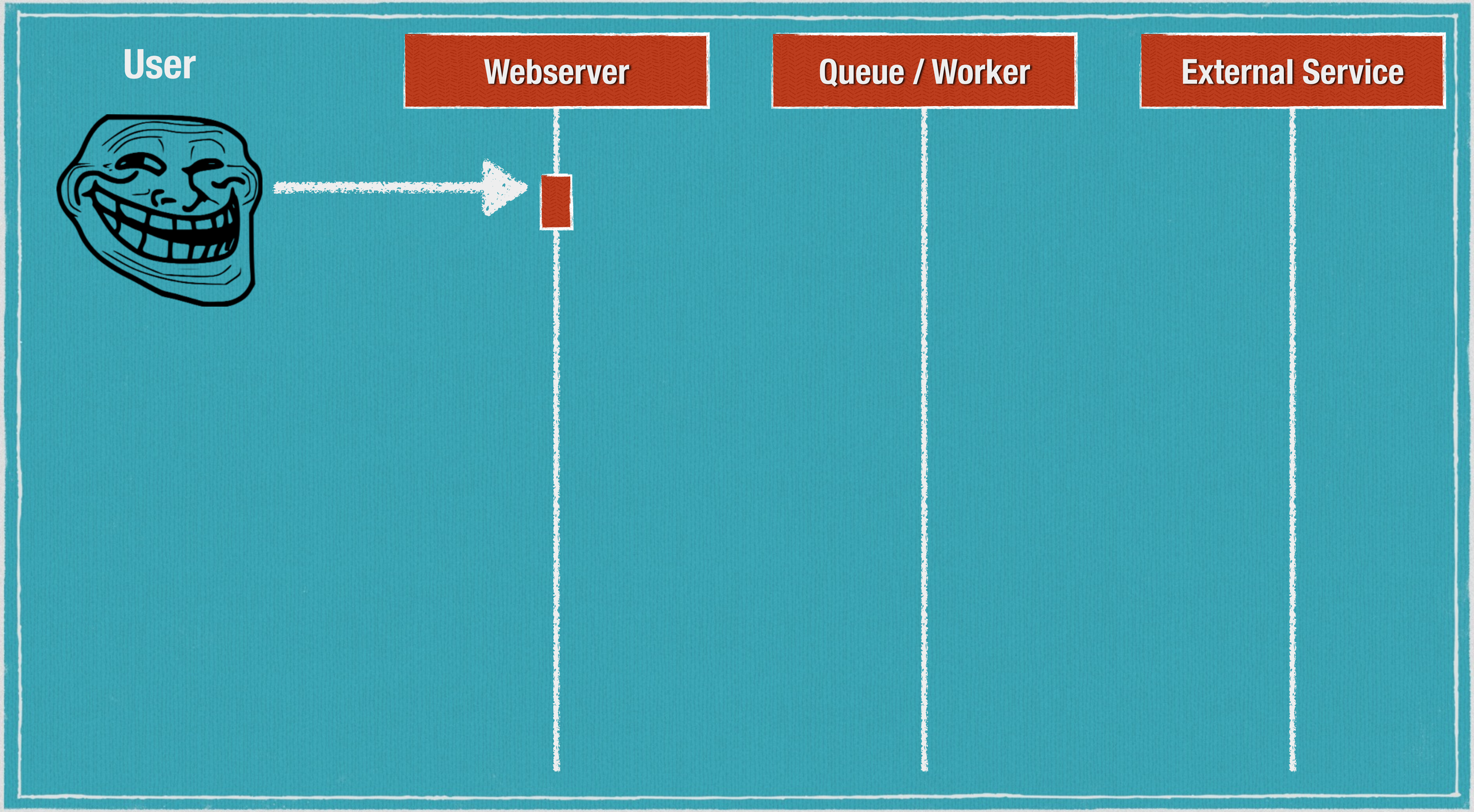
External Service

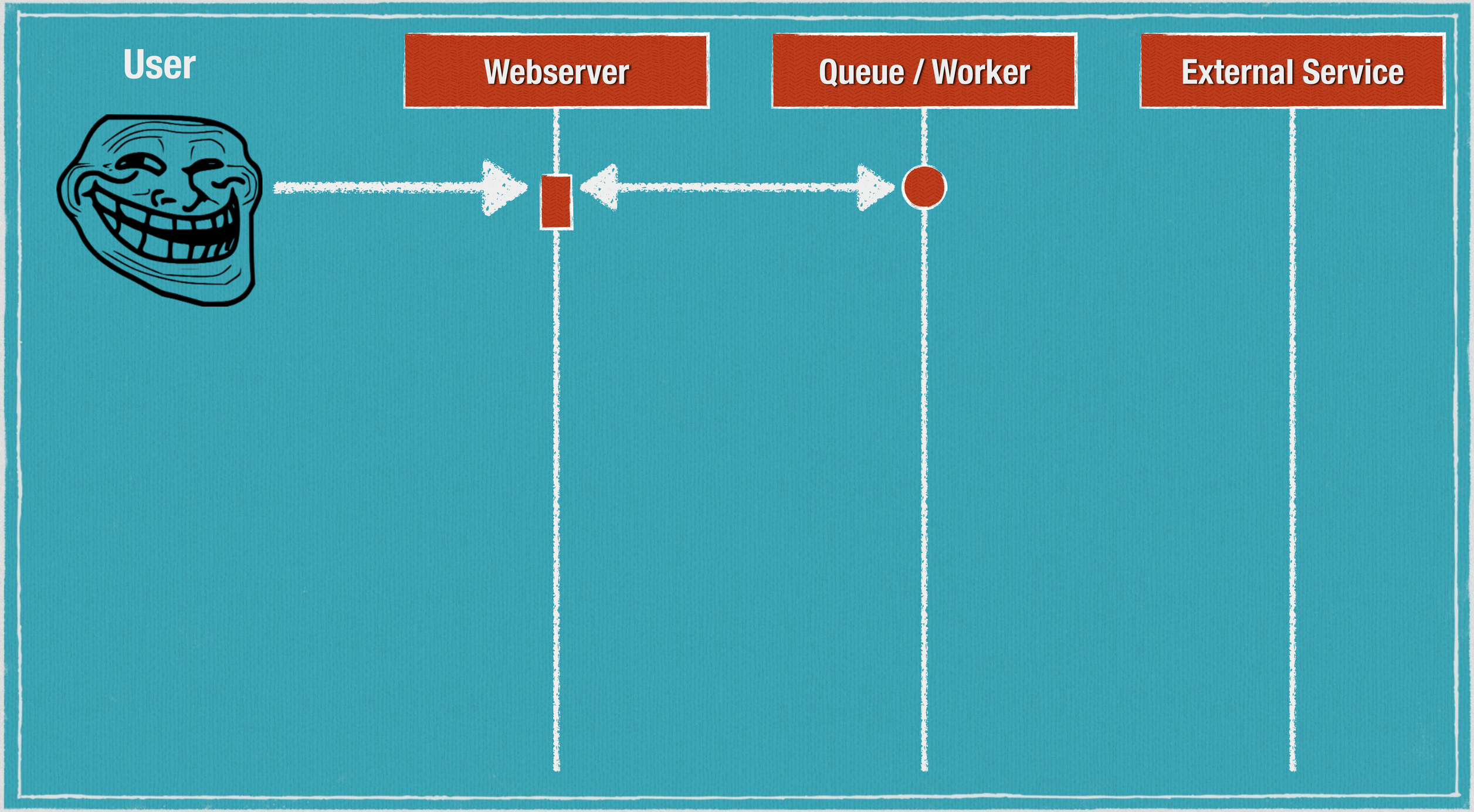
User

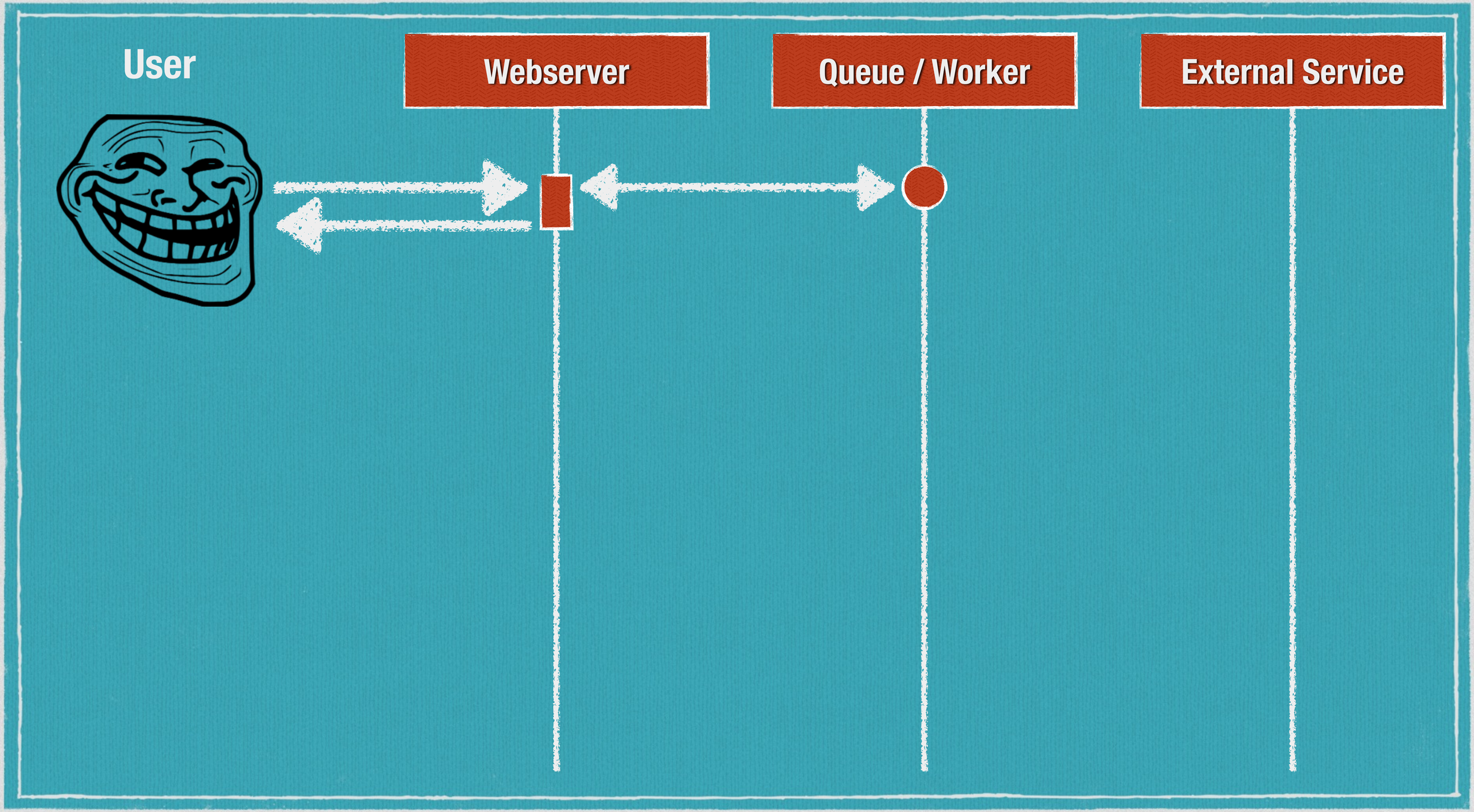
Webserver

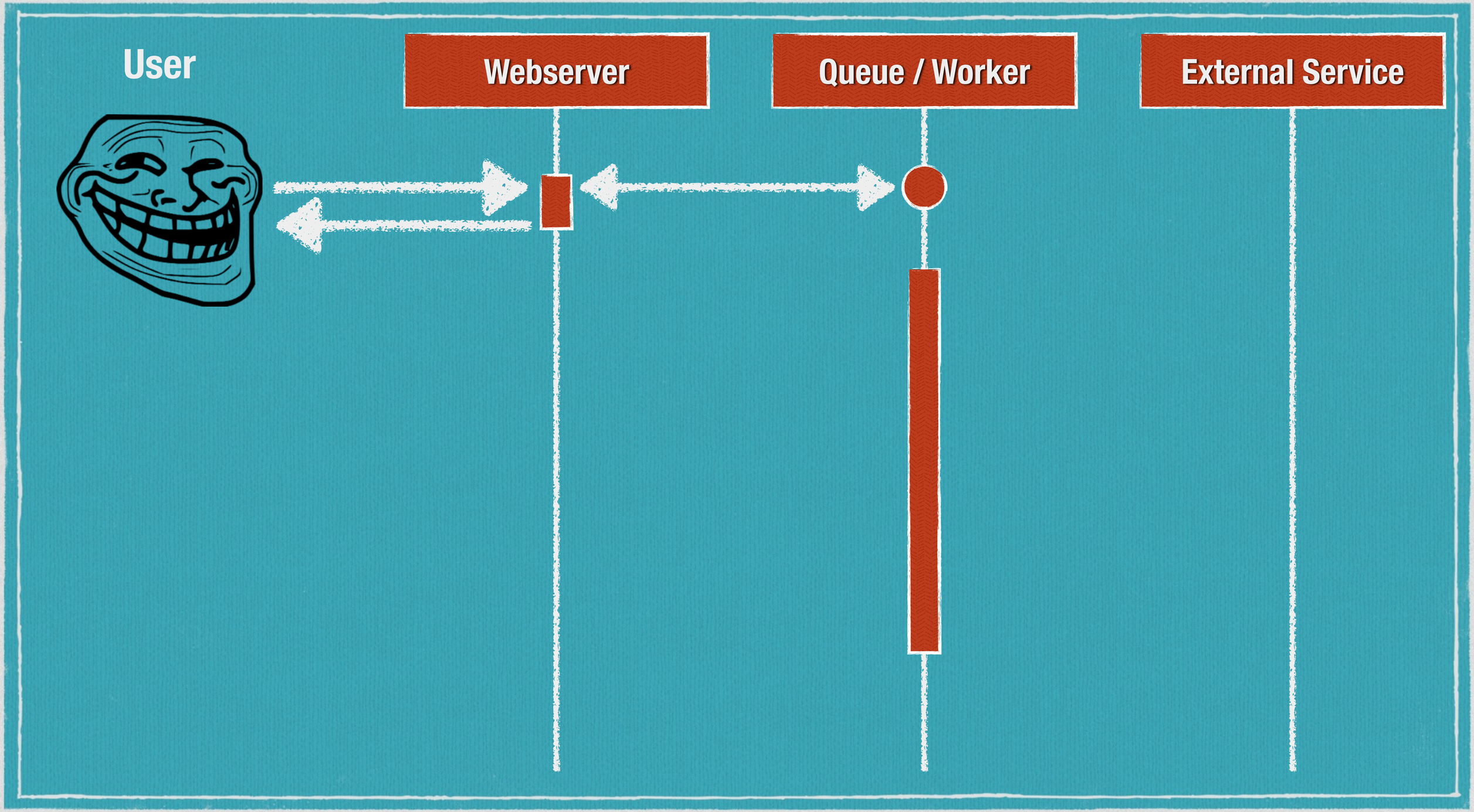
External Service

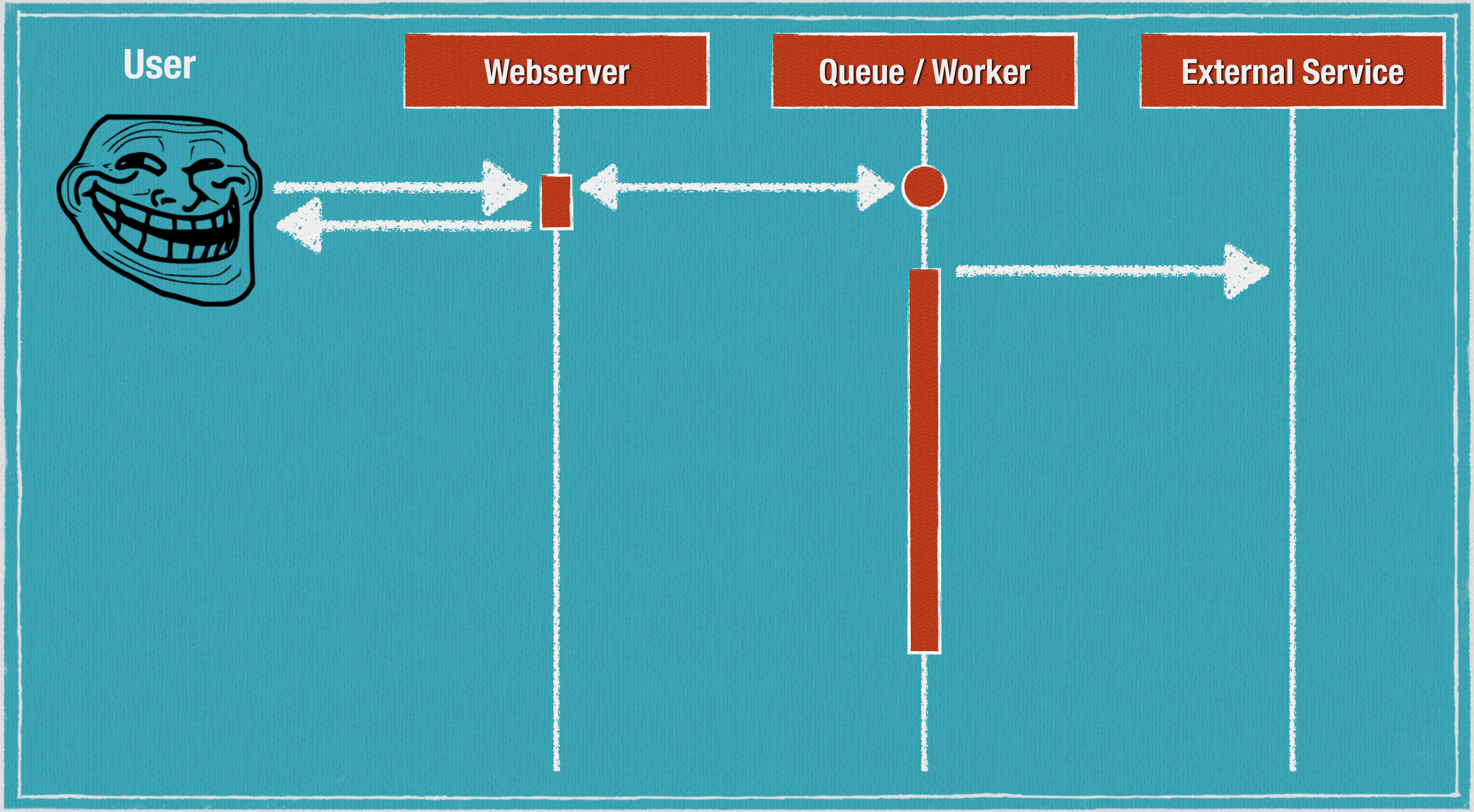

User
Webserver
External Service

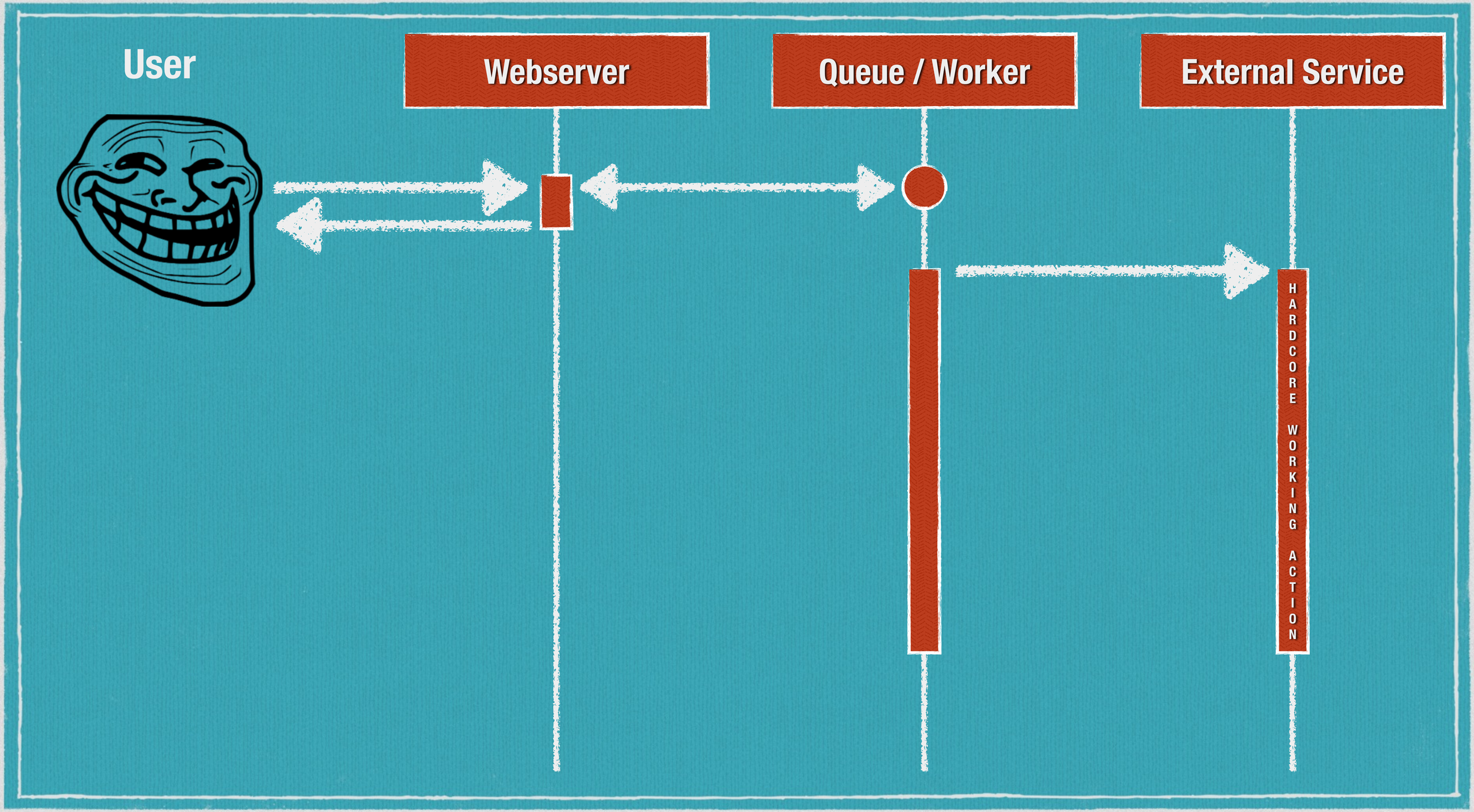

User
Webserver
Queue / Worker
External Service

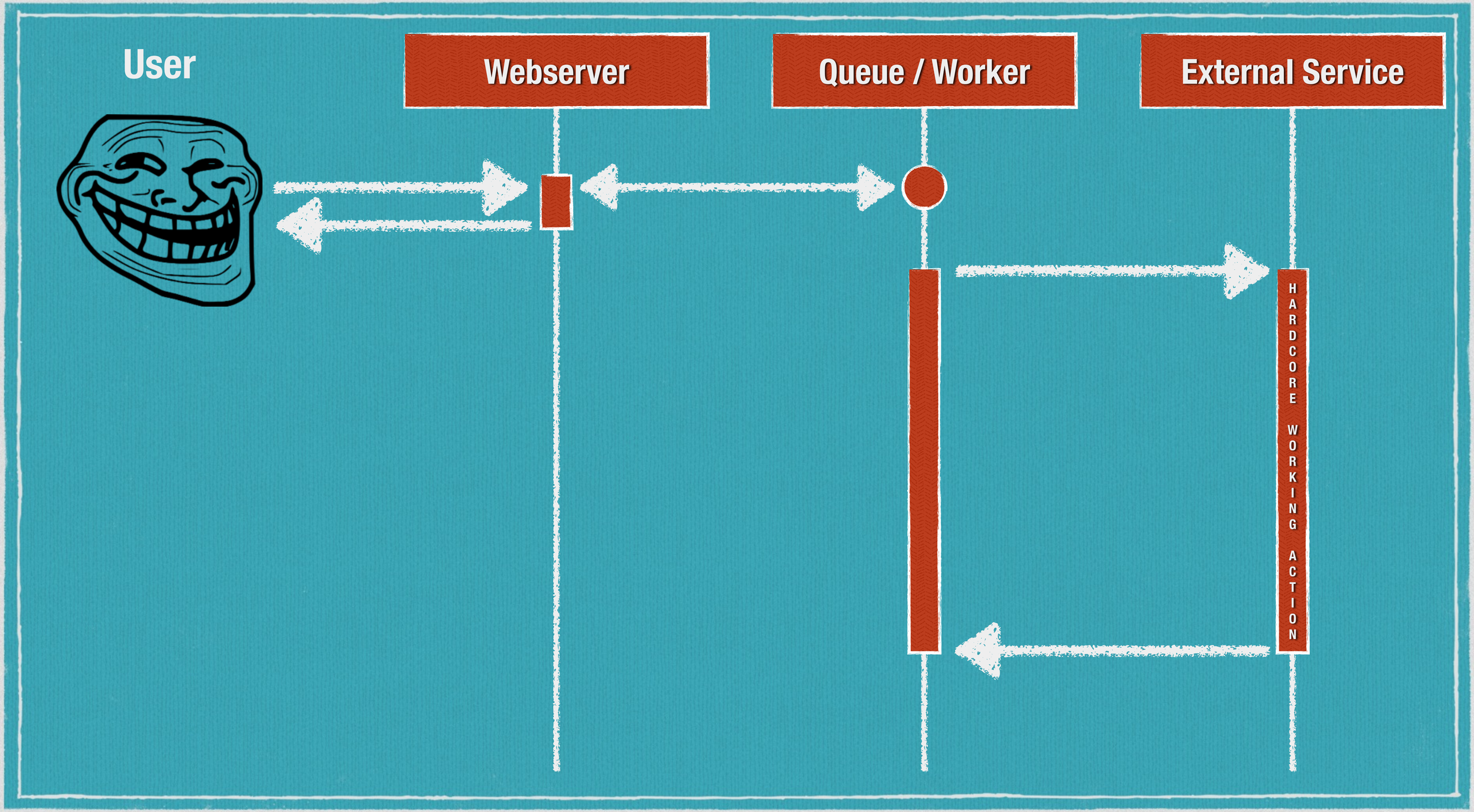

User
Webserver
Queue / Worker
External Service

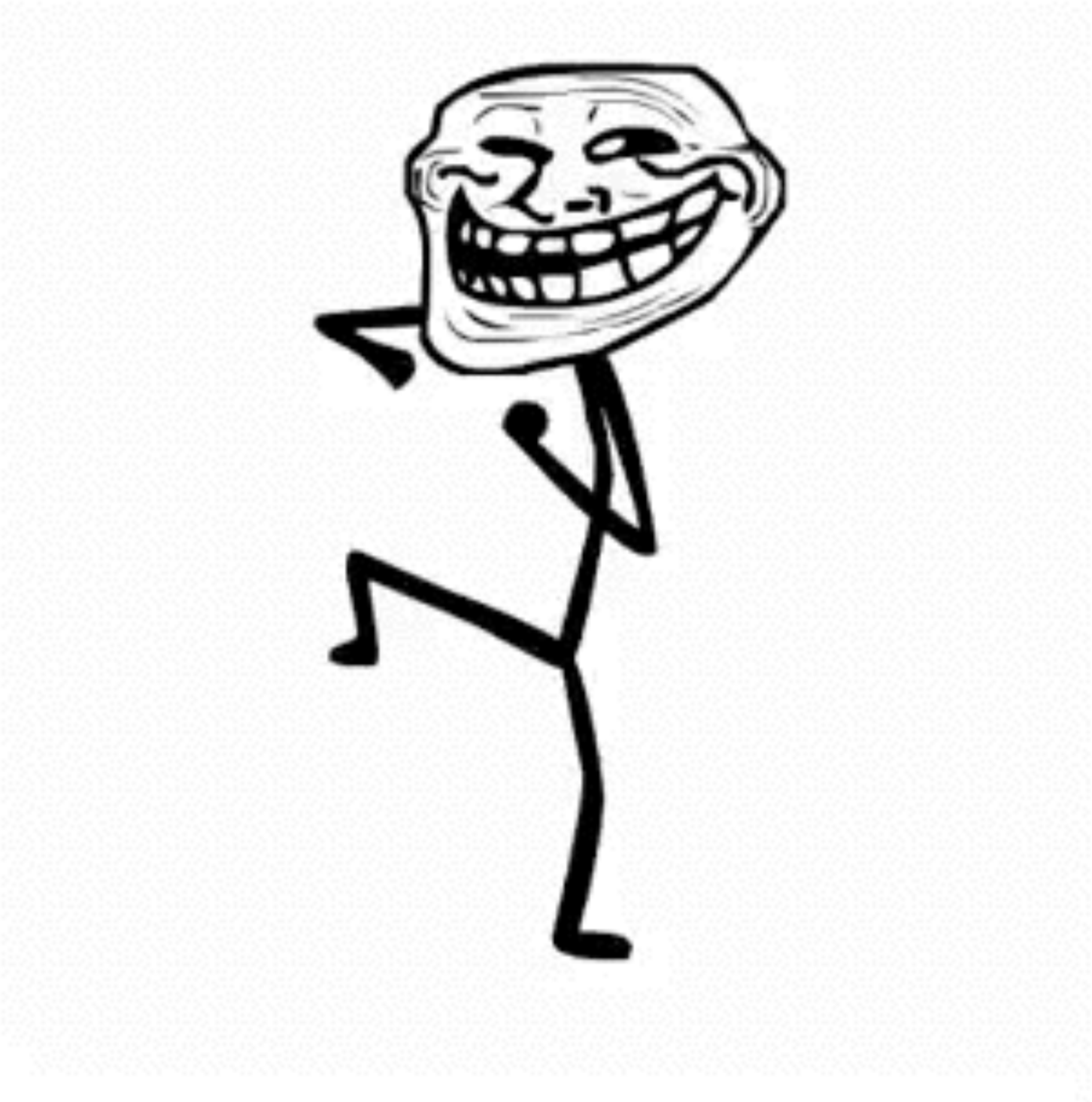

User
Webserver
Queue / Worker
External Service

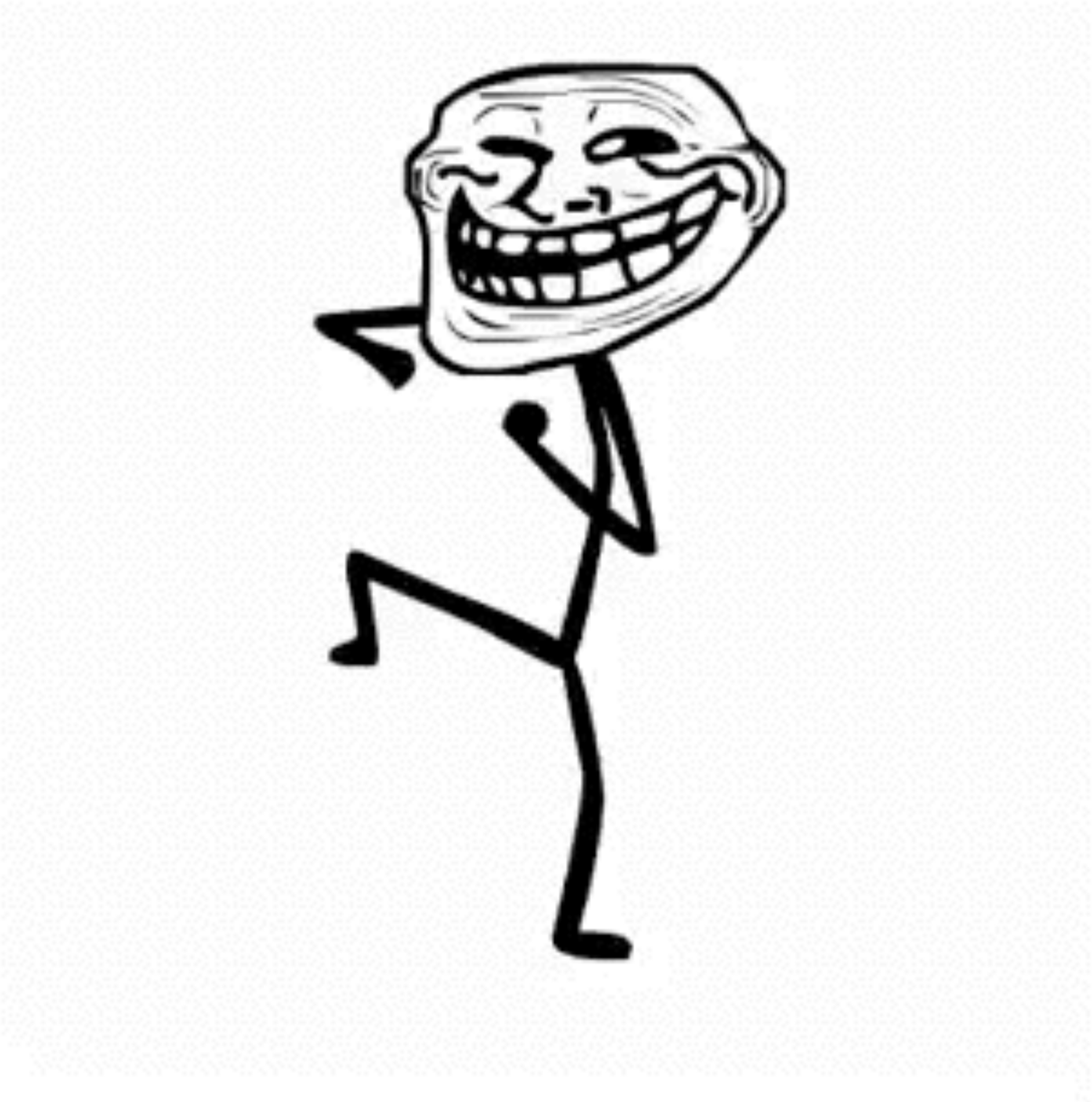

User
Webserver
Queue / Worker
External Service

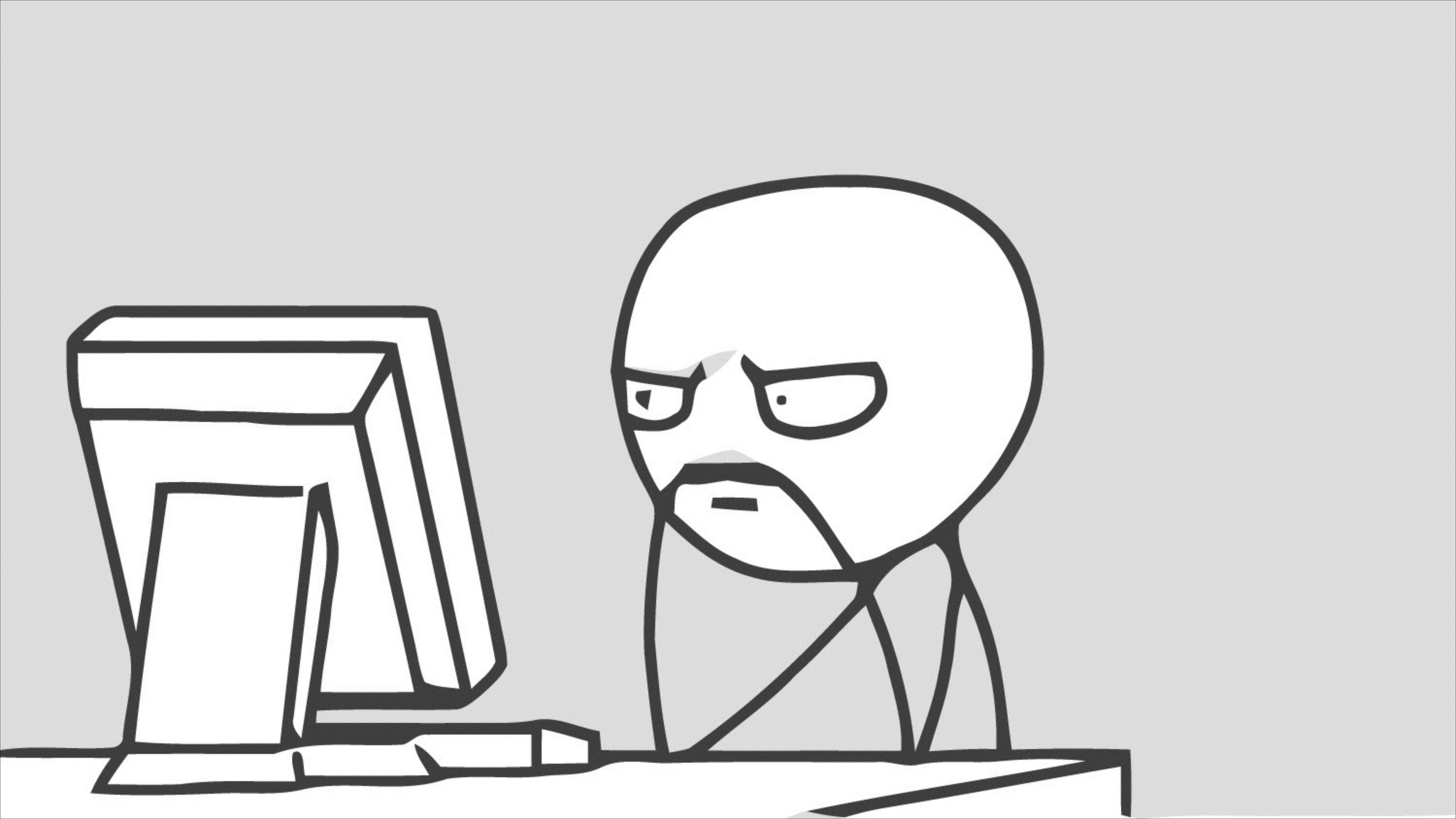

User
Webserver
Queue / Worker
External Service


User
Webserver
Queue / Worker
External Service
HARDCORE WORKING ACTION


User
Webserver
Queue / Worker
External Service
HARDCORE WORKING ACTION







it’s easy

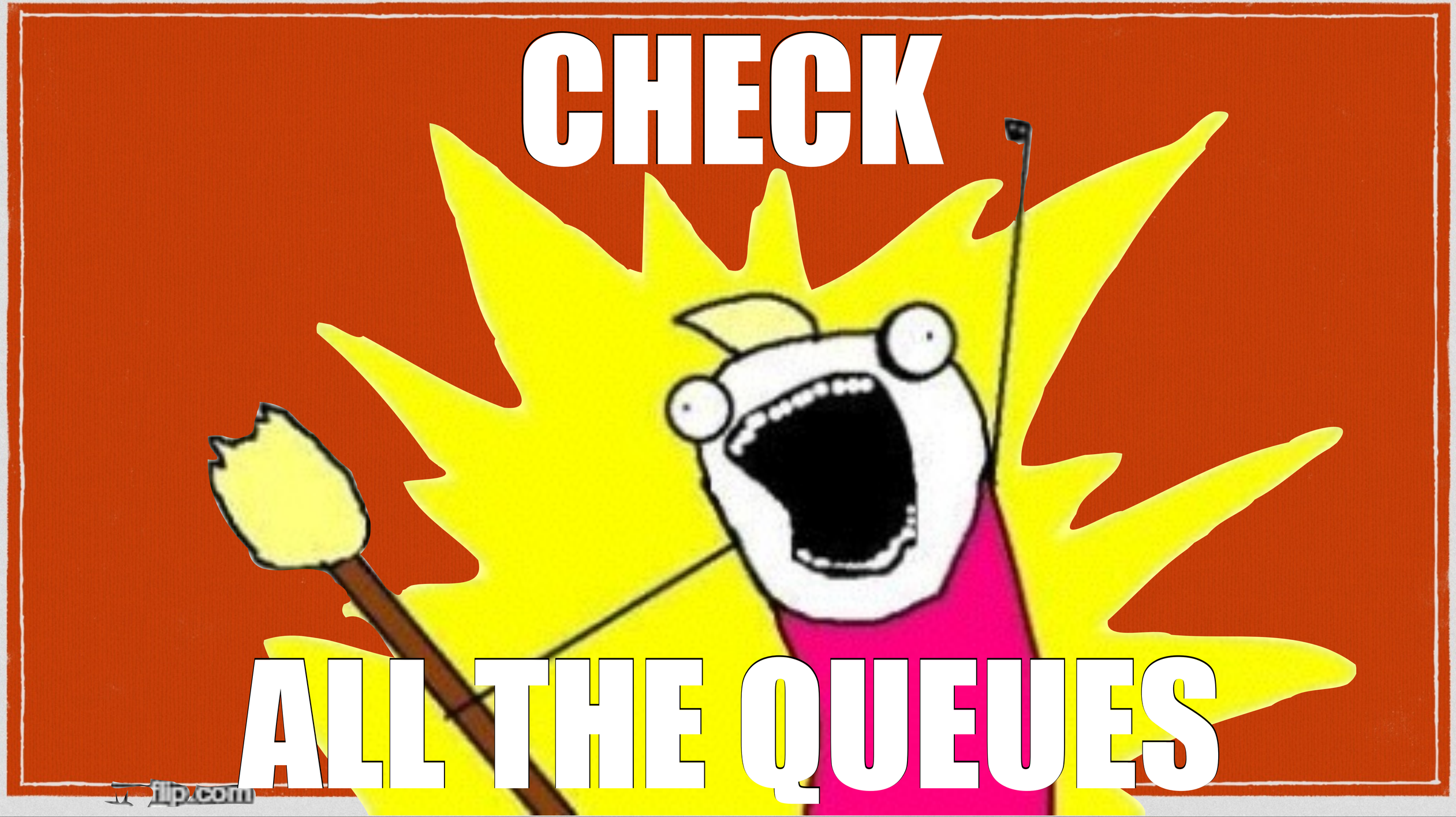
```
λ  simple-queue
```

```
λ  simple-queue
```

don’t try this at home

i4lcocl2 / shutterstock.com

REWIND!


CHECK
ALL THE QUEUES
flip.com

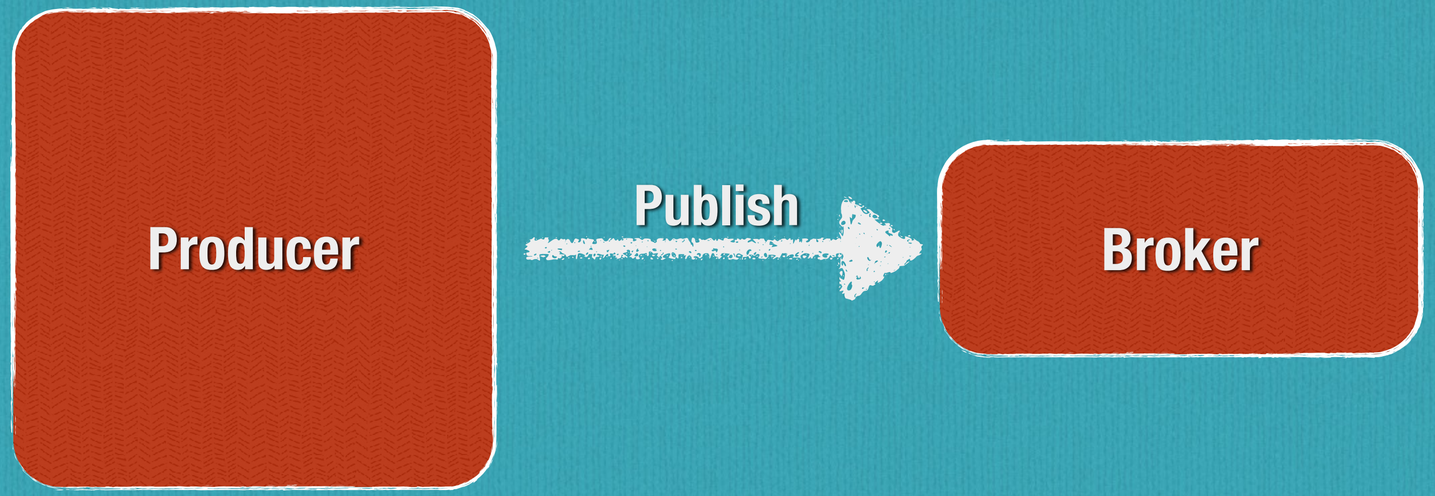
Message Queues

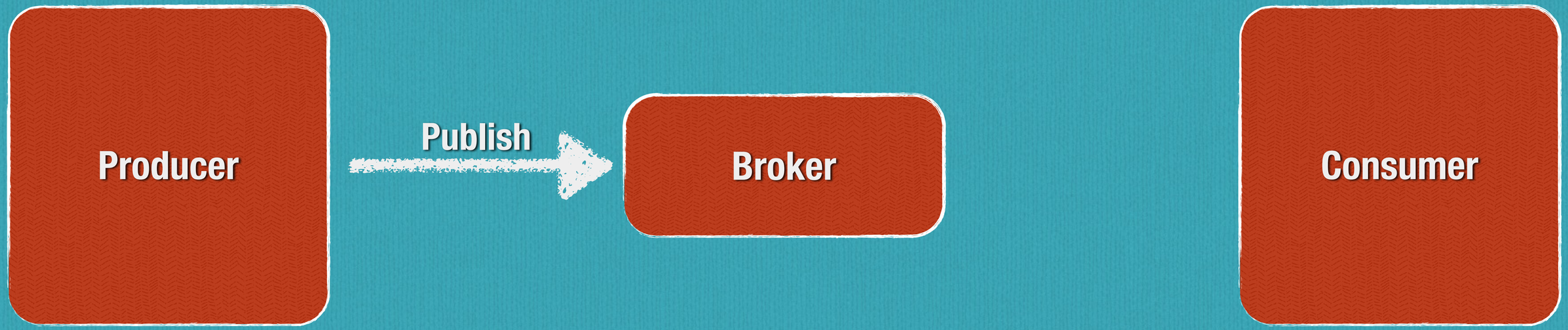
anatomies of a queue

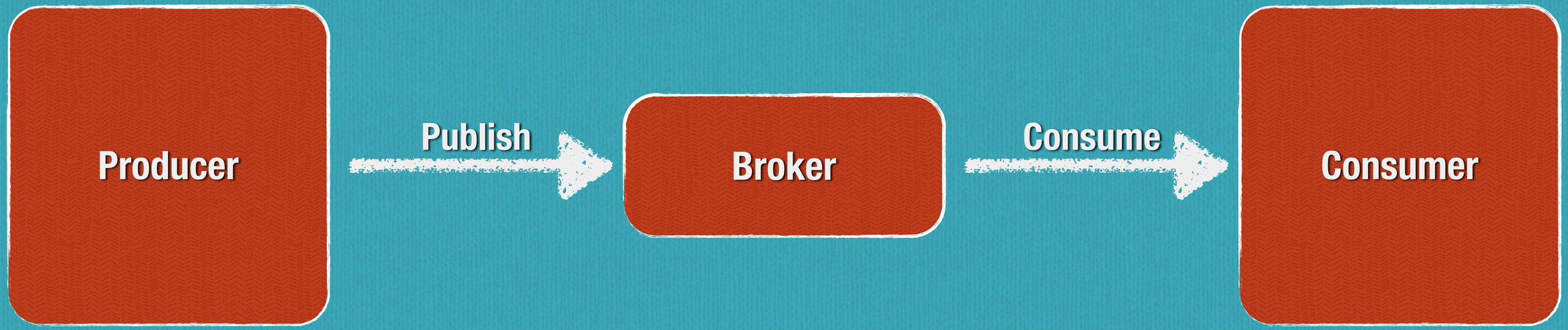
Producer

Producer

Broker

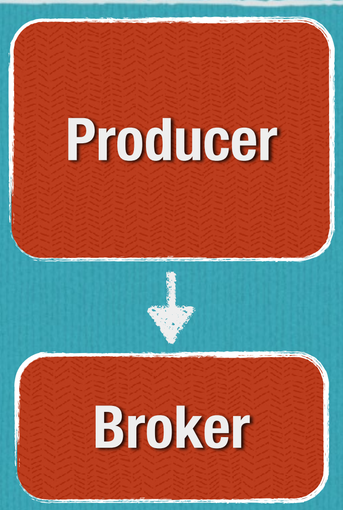

Producer
Publish
Broker

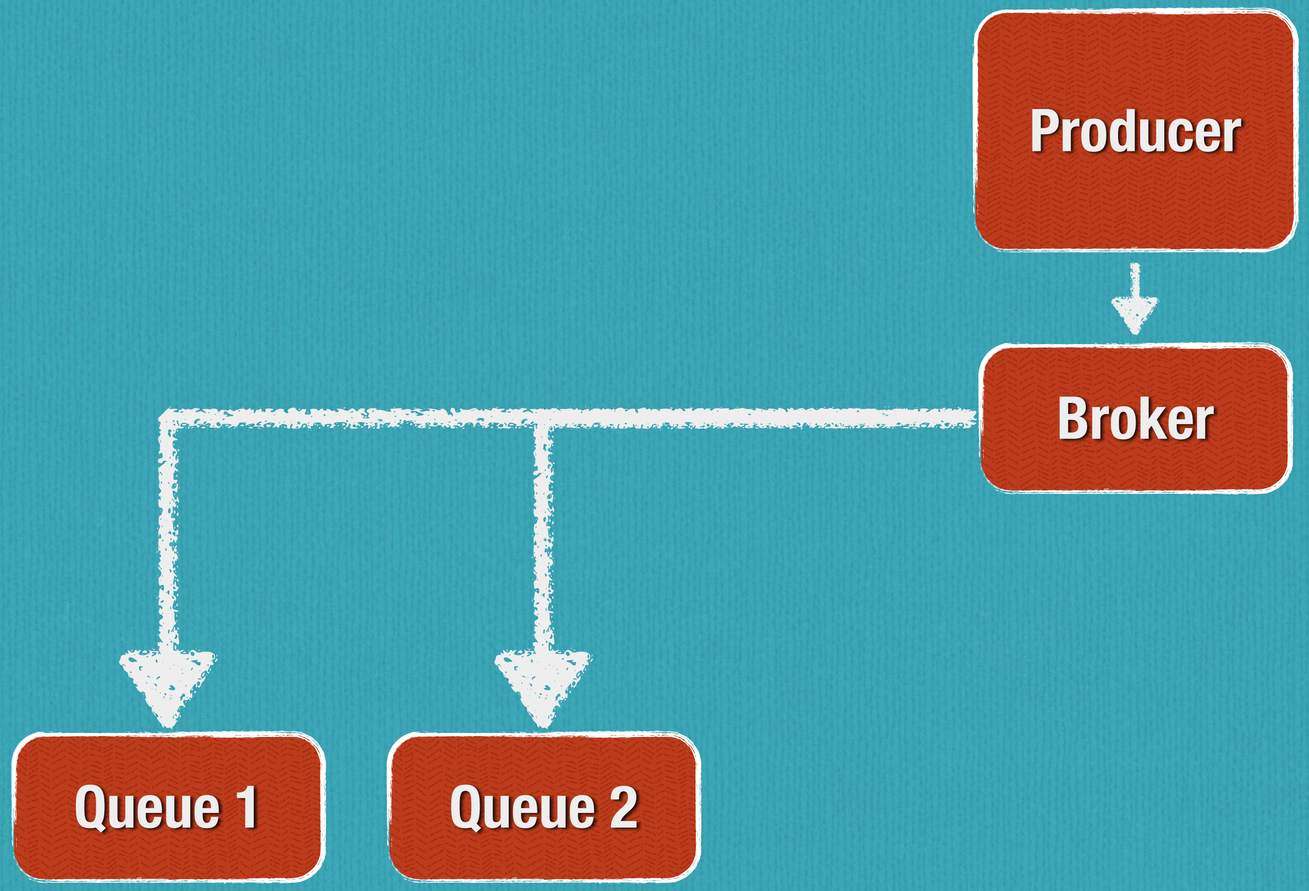

Producer
Publish
Broker
Consumer

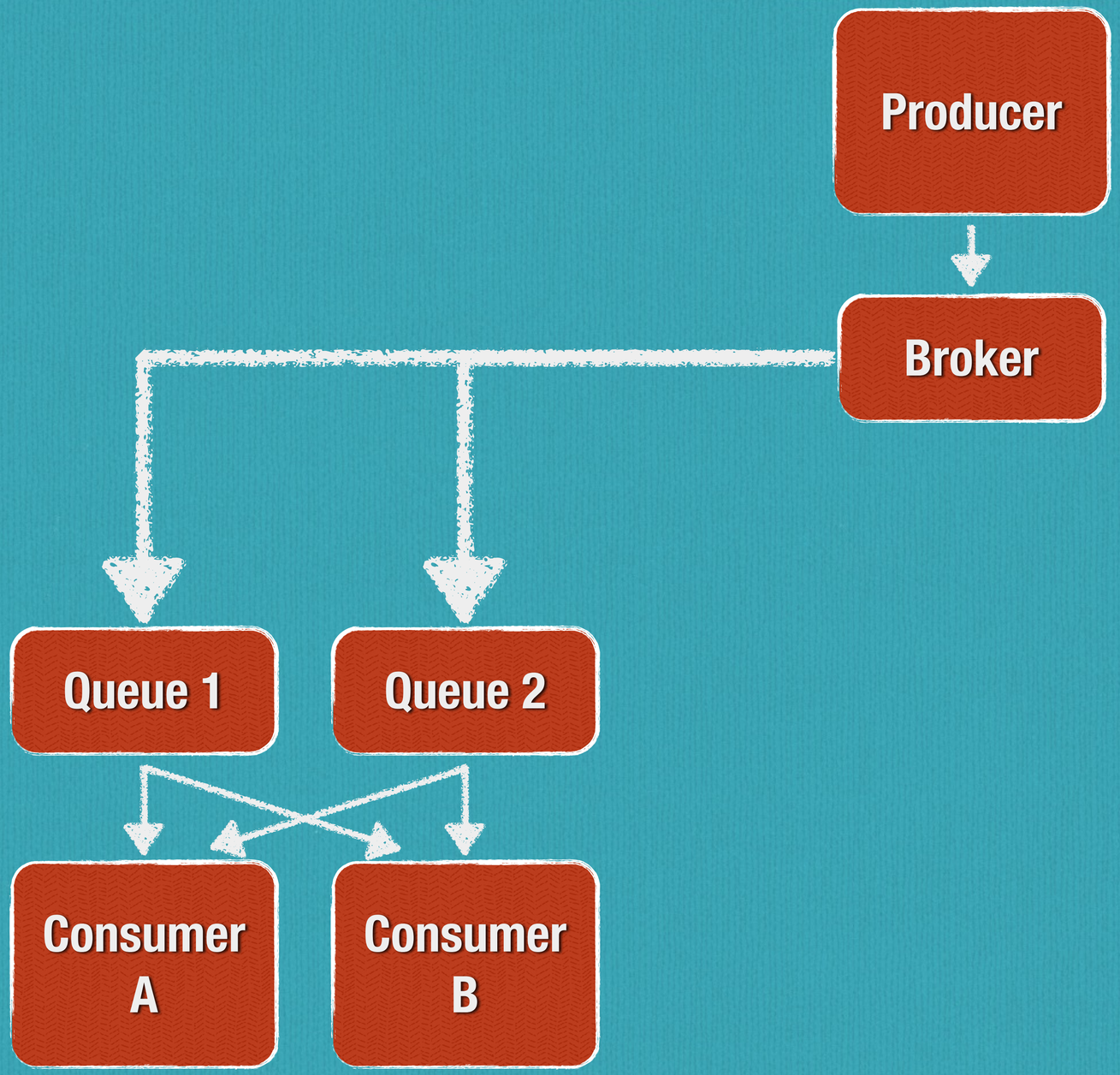

Producer
Publish
Broker
Consume
Consumer

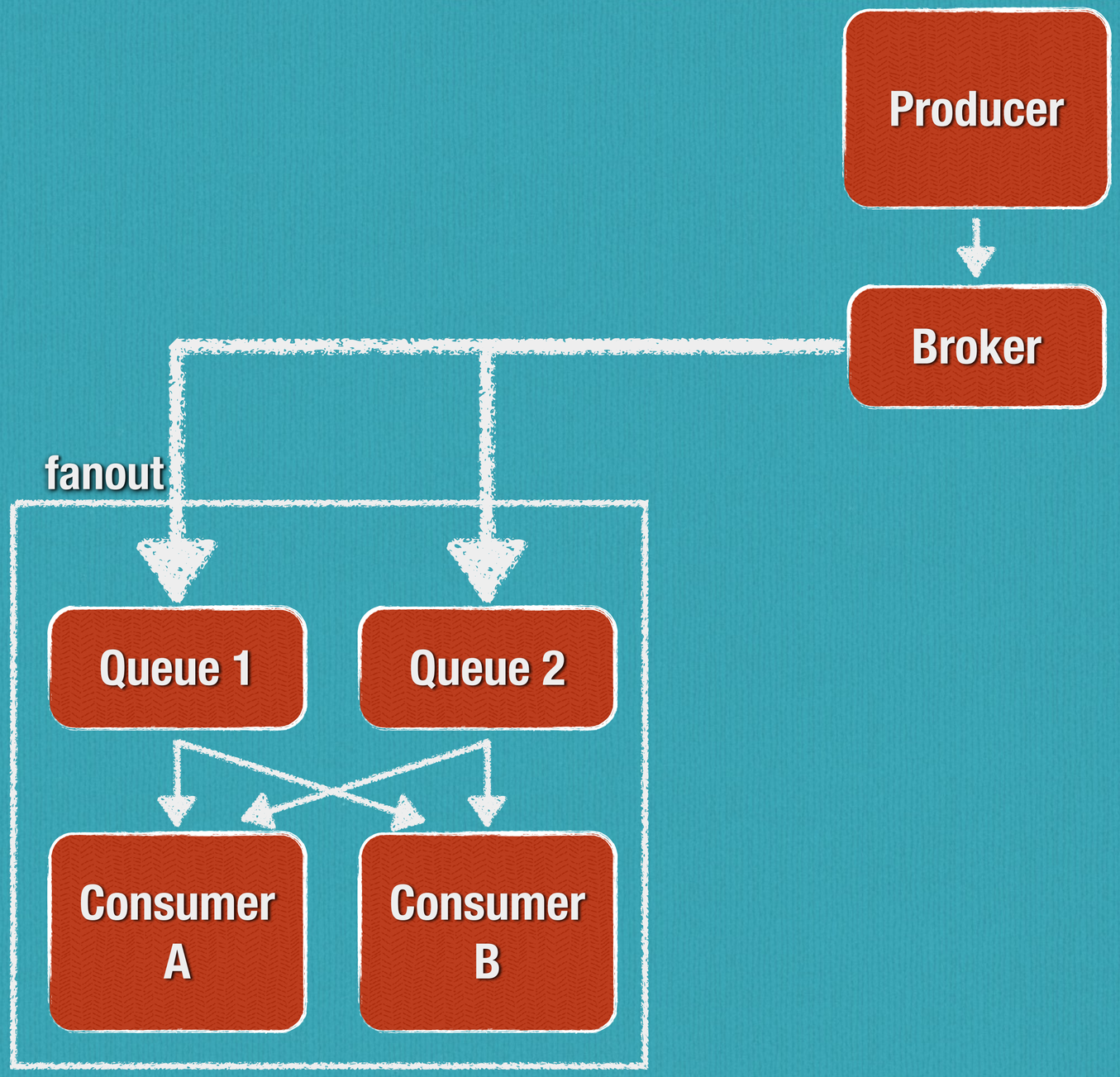
queue strategies

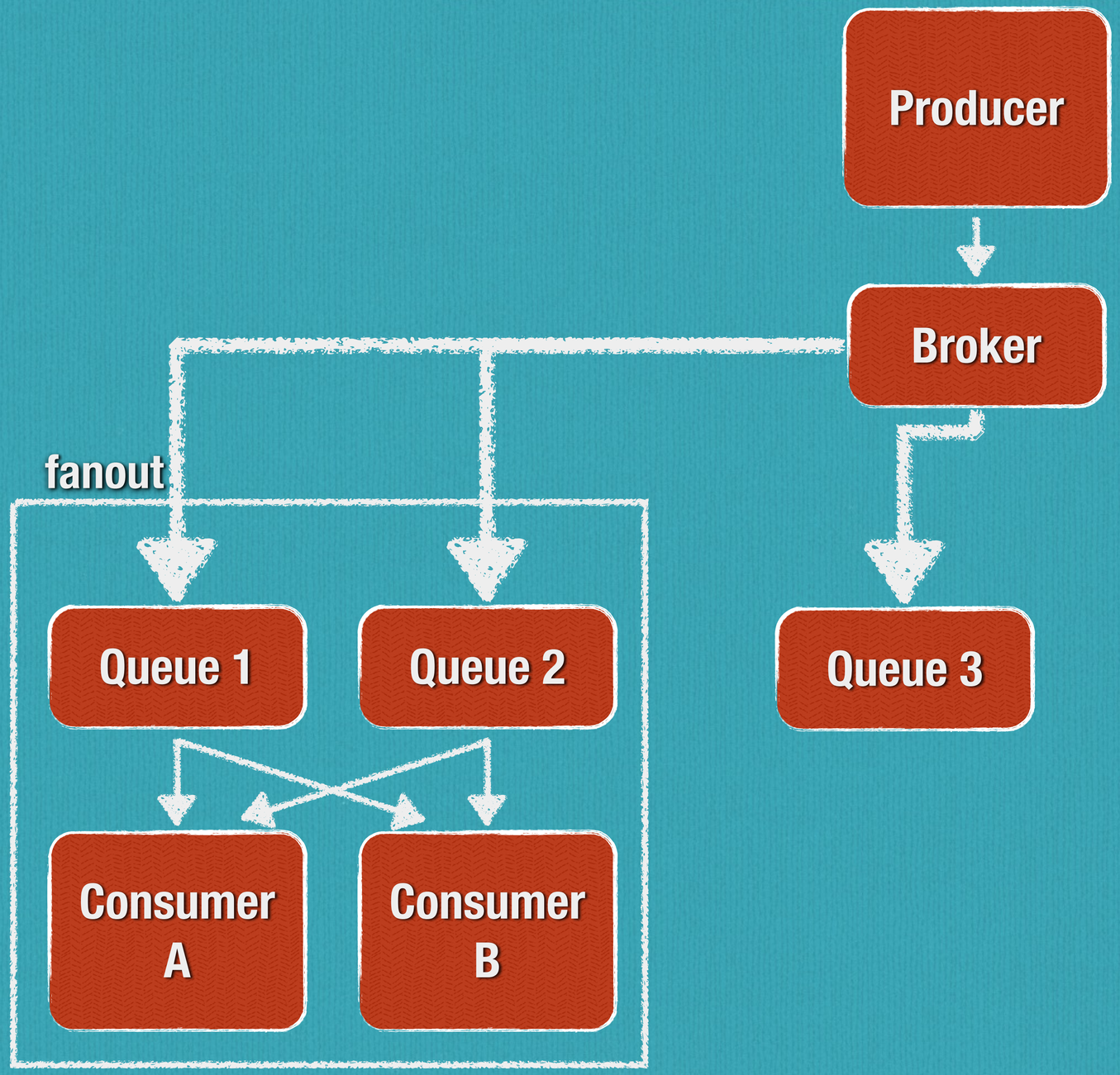
Producer

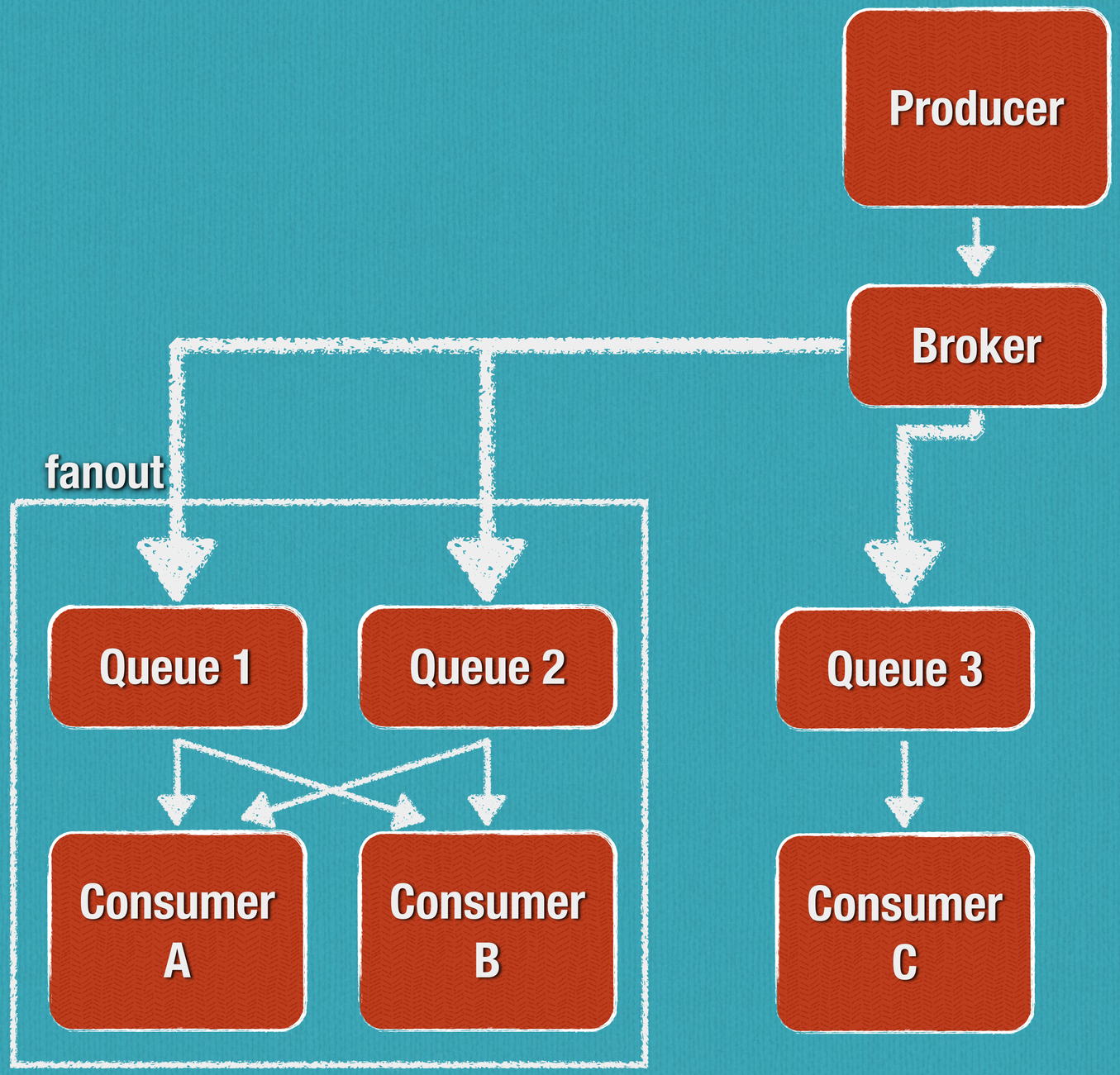

Producer
Broker

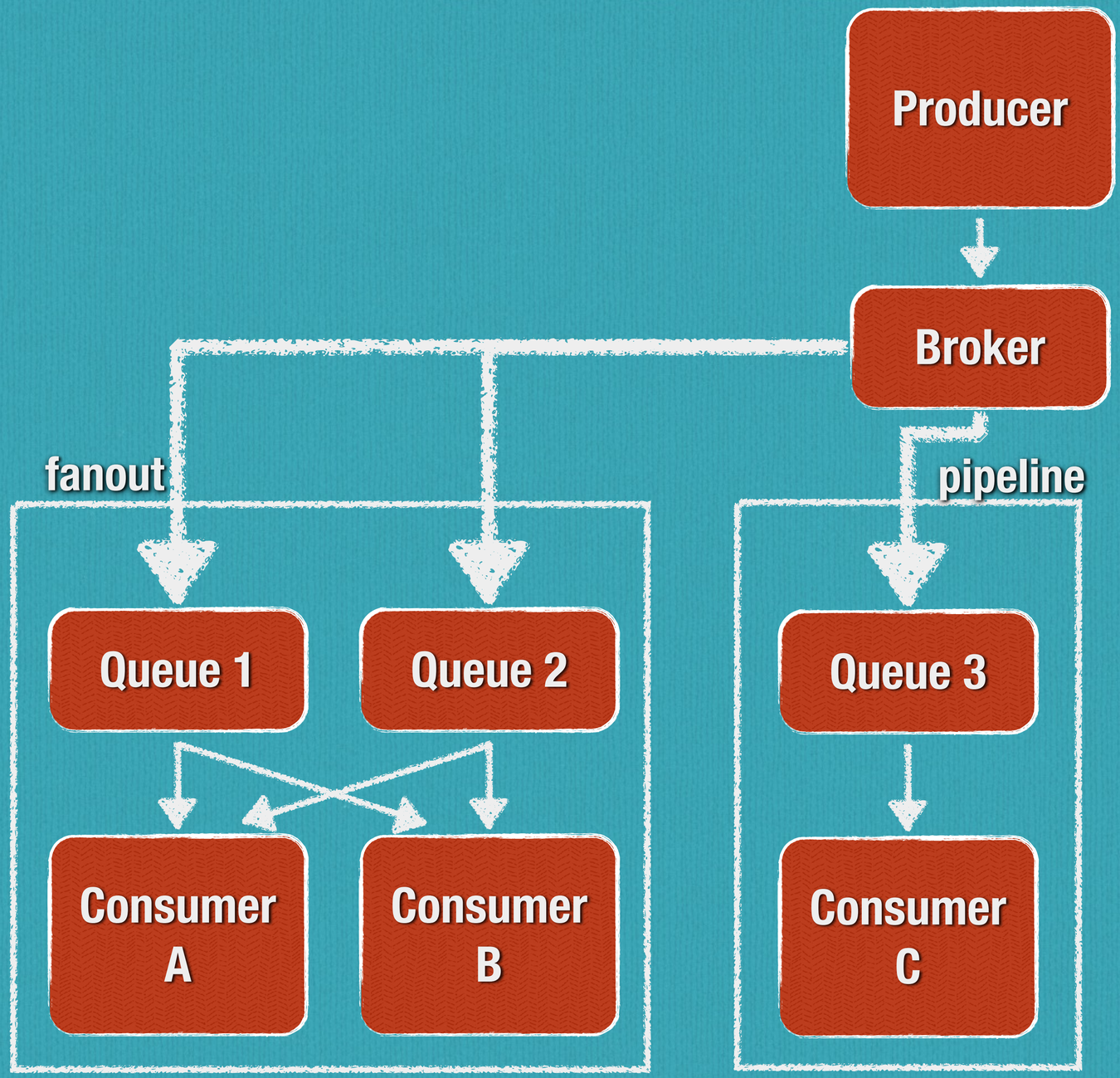

Producer
Broker
Queue 1
Queue 2

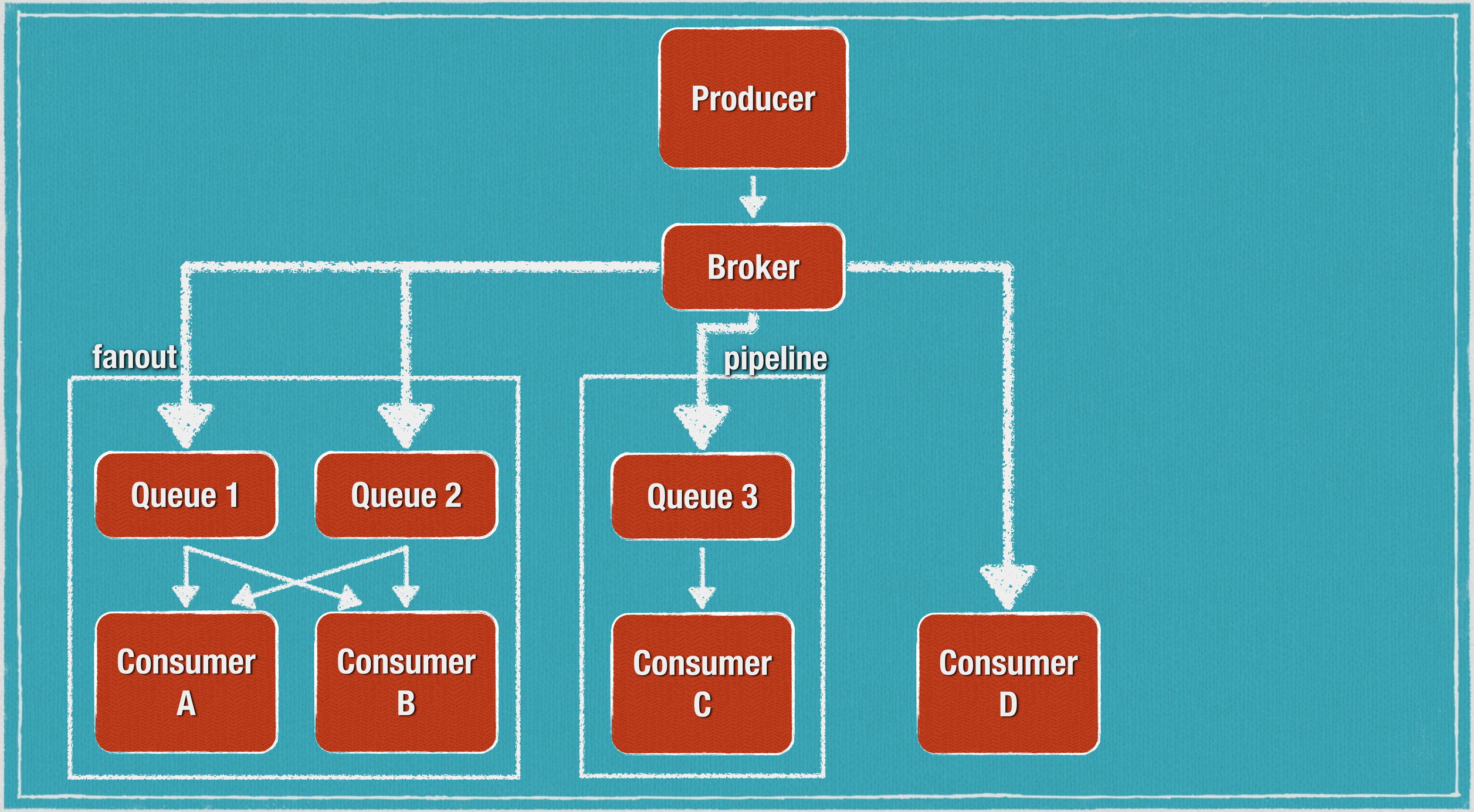

Producer
Broker
Queue 1
Queue 2
Consumer A
Consumer B

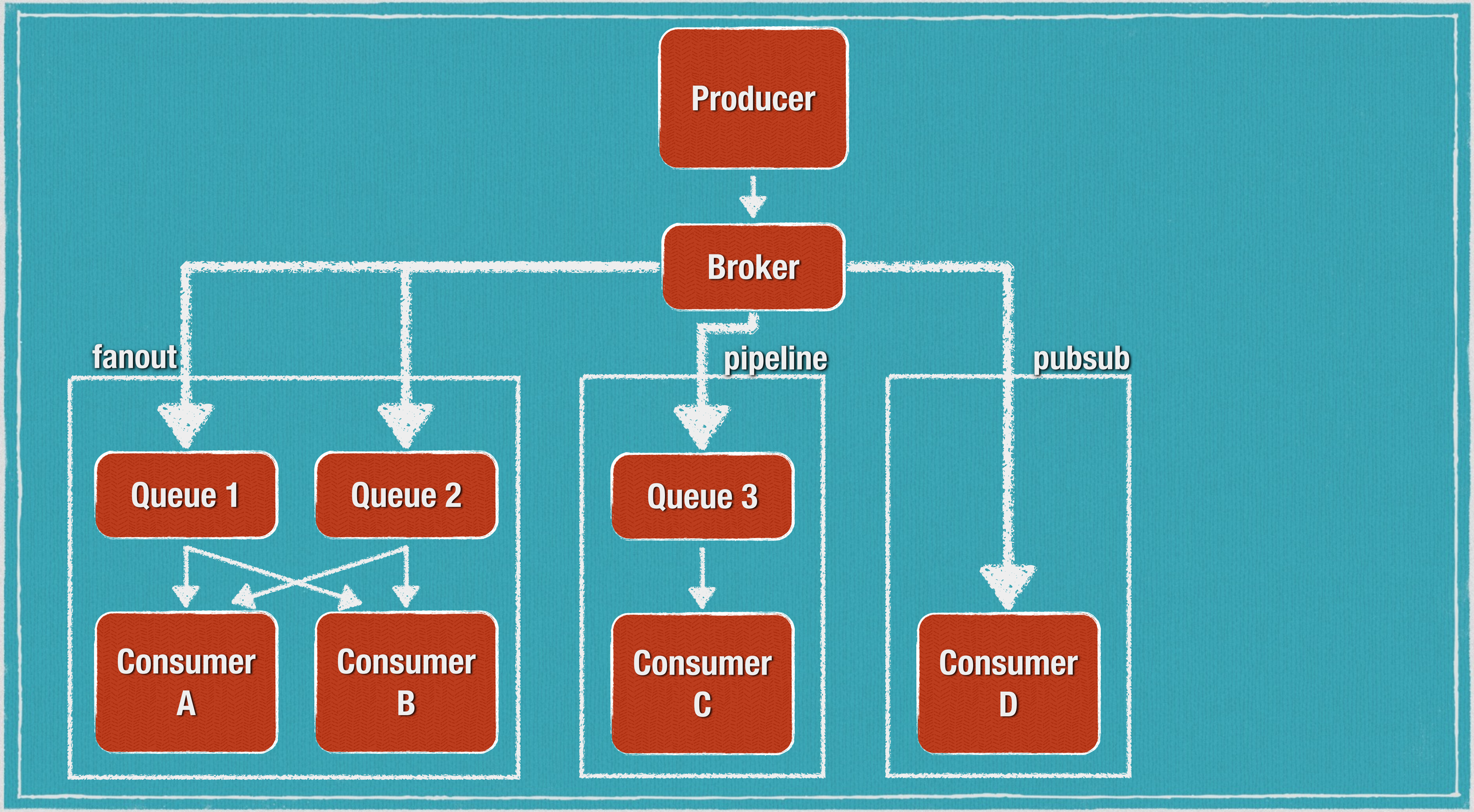

Producer
Broker
fanout
Queue 1
Queue 2
Consumer A
Consumer B

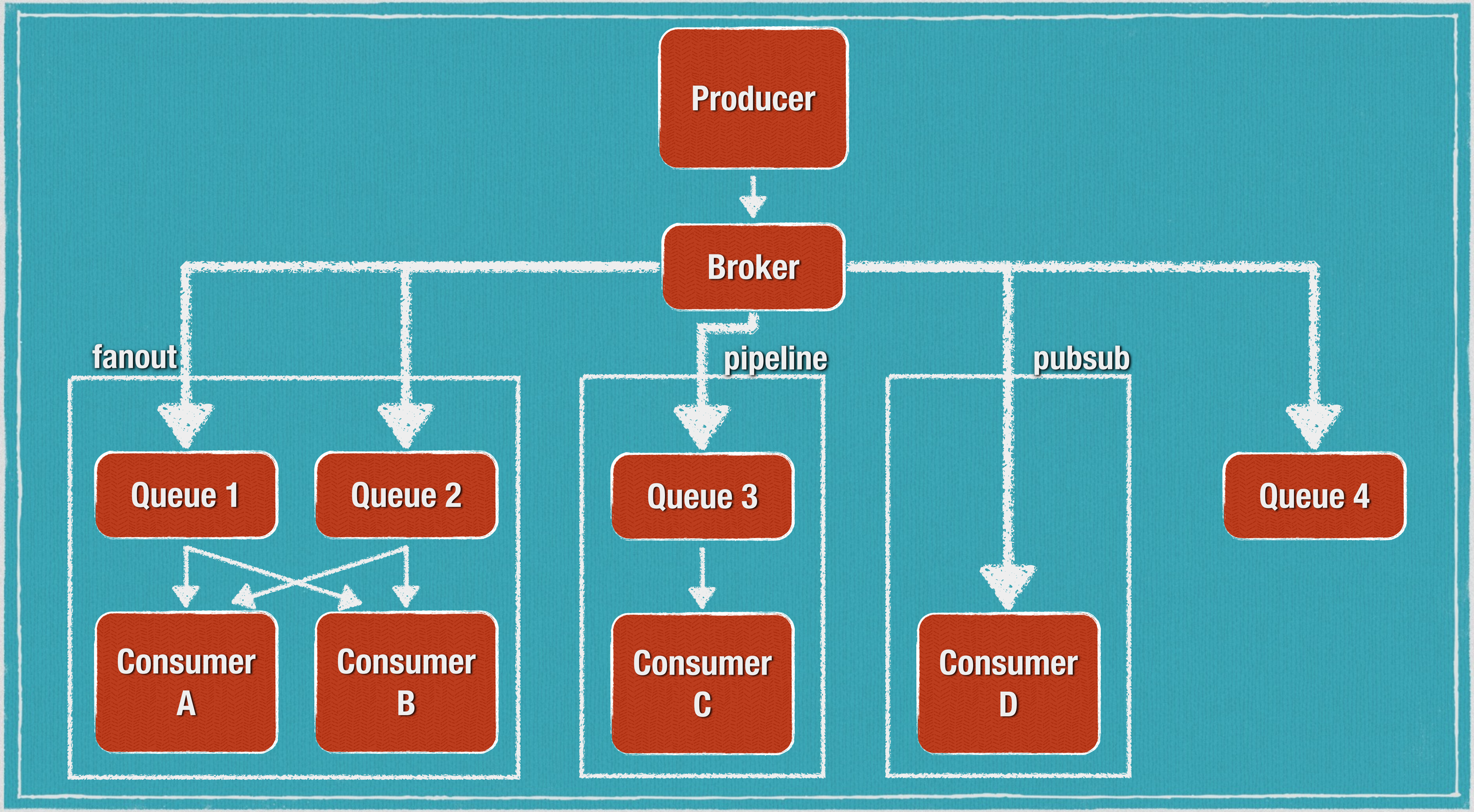

Producer
Broker
fanout
Queue 1
Queue 2
Queue 3
Consumer A
Consumer B

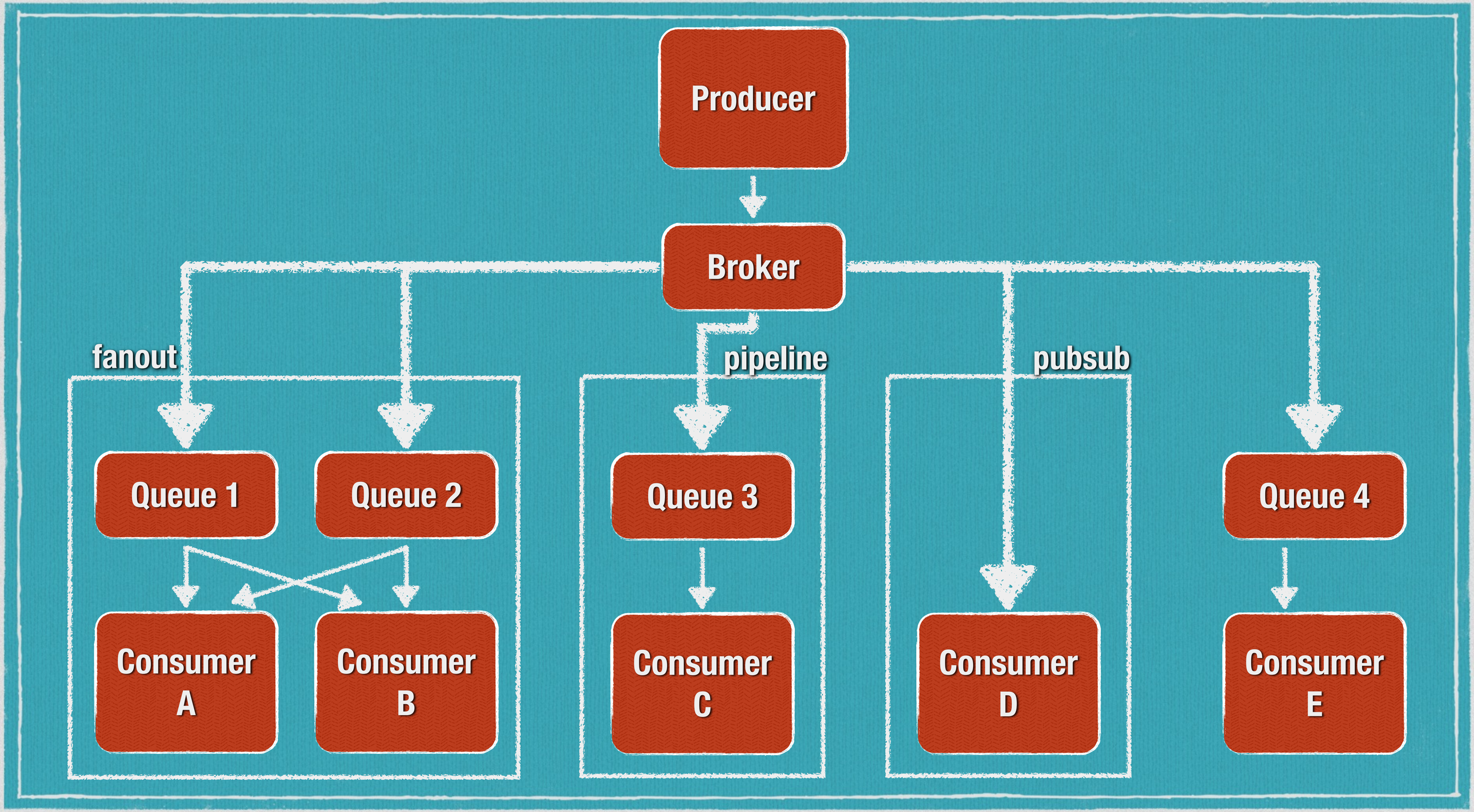

Producer
Broker
fanout
Queue 1
Queue 2
Queue 3
Consumer A
Consumer B
Consumer C

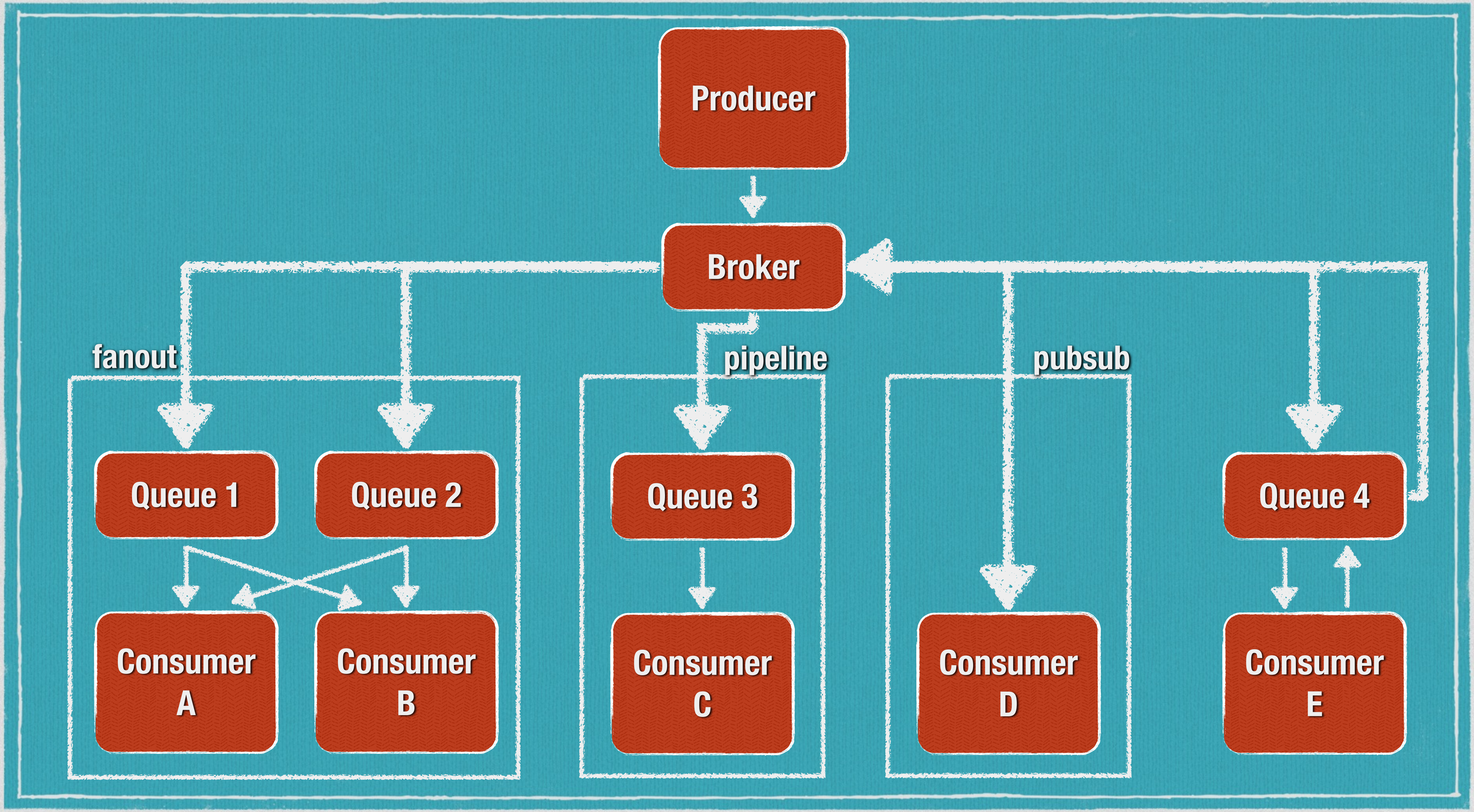

Producer
Broker
fanout
pipeline
Queue 1
Queue 2
Queue 3
Consumer A
Consumer B
Consumer C

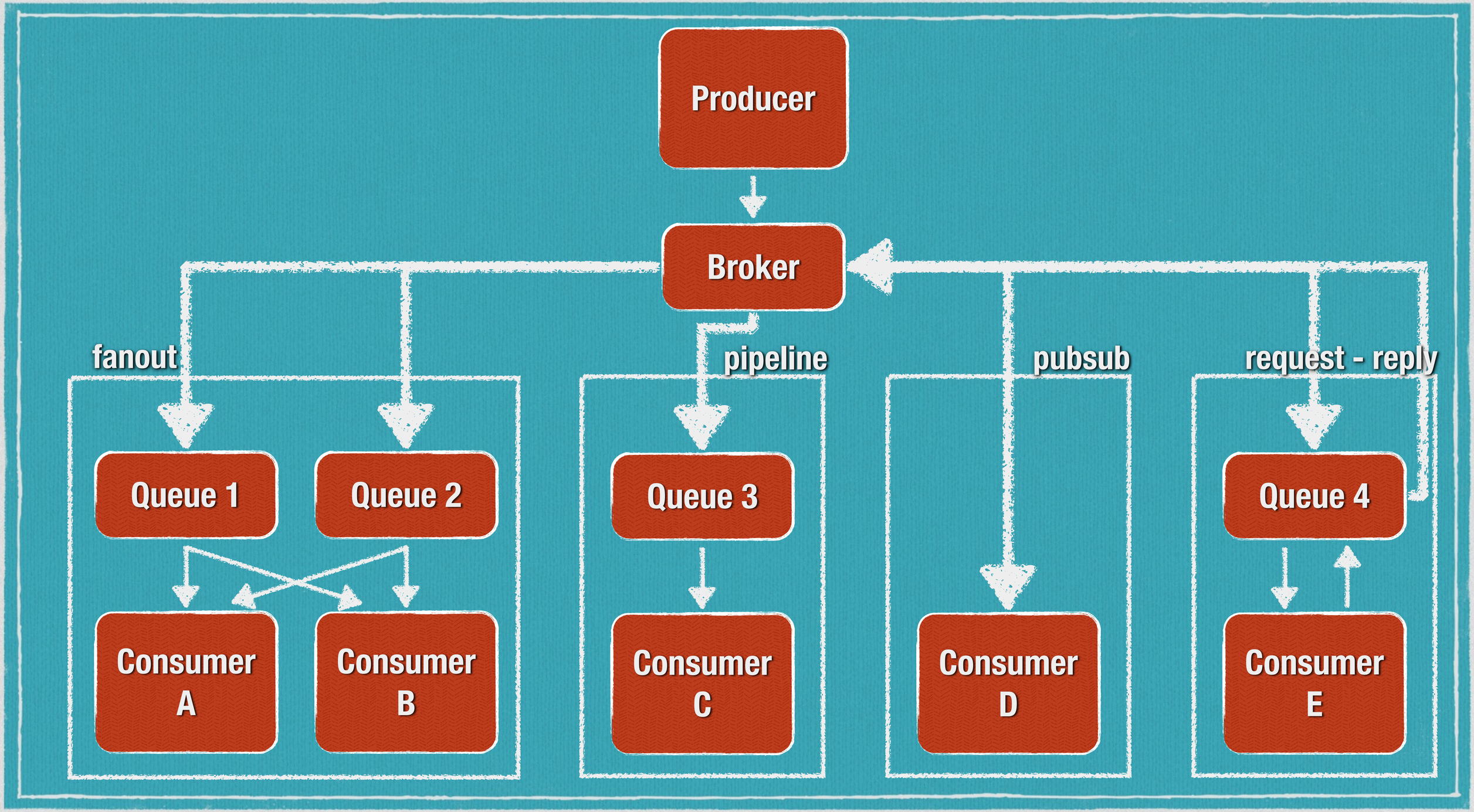

Producer
Broker
fanout
pipeline
Queue 1
Queue 2
Queue 3
Consumer A
Consumer B
Consumer C
Consumer D


Producer
Broker
fanout
pipeline
pubsub
Queue 1
Queue 2
Queue 3
Consumer A
Consumer B
Consumer C
Consumer D


Producer
Broker
fanout
pipeline
pubsub
Queue 1
Queue 2
Queue 3
Queue 4
Consumer A
Consumer B
Consumer C
Consumer D


Producer
Broker
fanout
pipeline
pubsub
Queue 1
Queue 2
Queue 3
Queue 4
Consumer A
Consumer B
Consumer C
Consumer D
Consumer E


Producer
Broker
fanout
pipeline
pubsub
Queue 1
Queue 2
Queue 3
Queue 4
Consumer A
Consumer B
Consumer C
Consumer D
Consumer E


Producer
Broker
fanout
pipeline
pubsub
request - reply
Queue 1
Queue 2
Queue 3
Queue 4
Consumer A
Consumer B
Consumer C
Consumer D
Consumer E

feature check

message distribution

broadcasting

Job Queues

designed for jobs

feature check

persistence

priorities

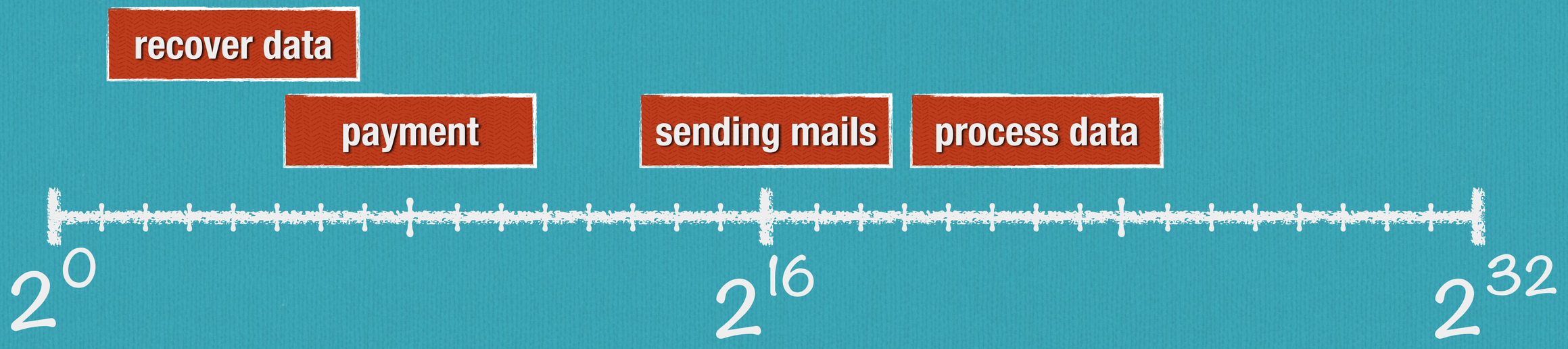
priorities

priorities

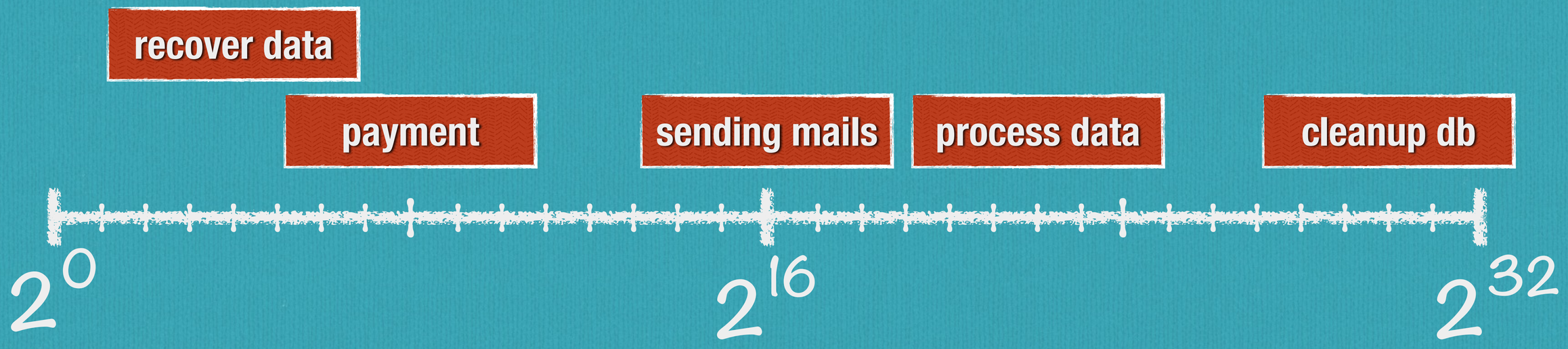
priorities

priorities

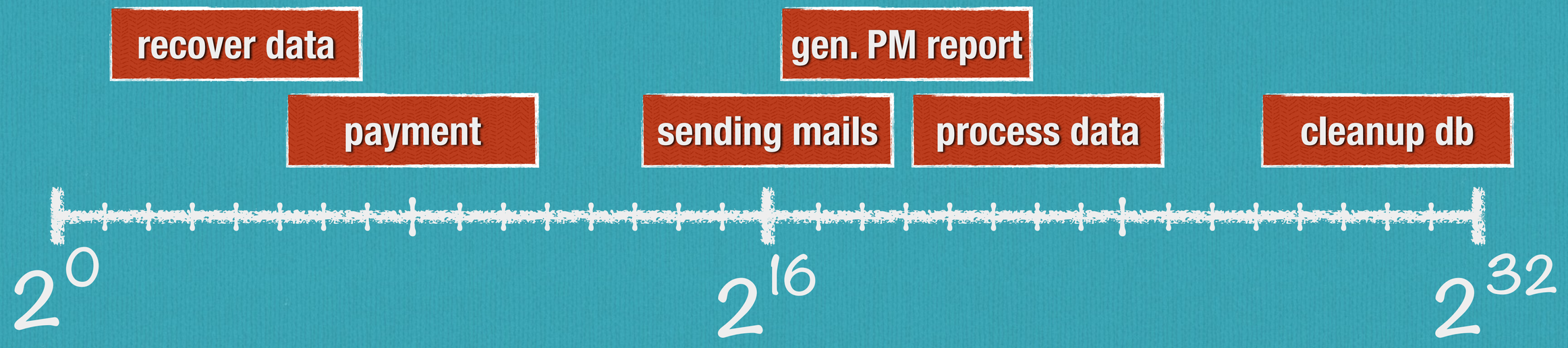
priorities



priorities


recover data
payment
sending mails
process data
cleanup db
2^0
2^{16}
2^{32}

priorities

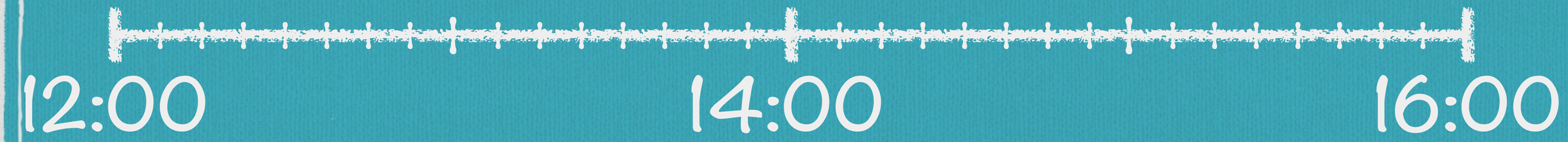

recover data
payment
gen. PM report
sending mails
process data
cleanup db
2^0
2^{16}
2^{32}

priorities

recover data
gen. PM report
payment
sending mails
process data
cleanup db
2^0
2^{16}
2^{32}

timing

timing - job delay

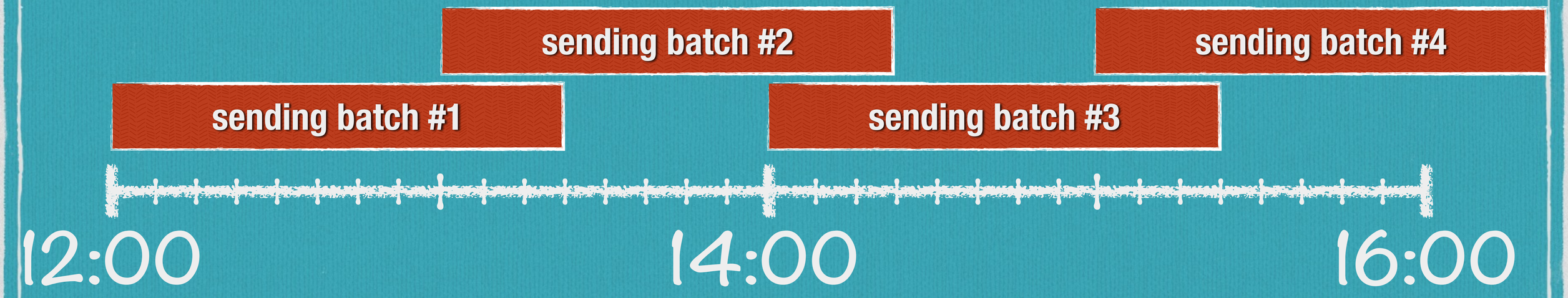


timing - job delay

timing - job delay

timing - job delay

timing - job delay



What Queue Should I Use?

USE WHATEVER MAKES FUCKING SENSE!

But We Can Check A Few

But Ca Few

allrounders

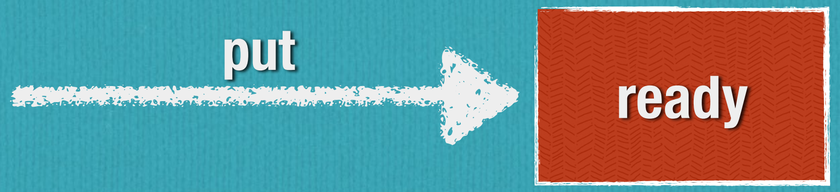
SaaS

no queues

job queues

beanstalkd

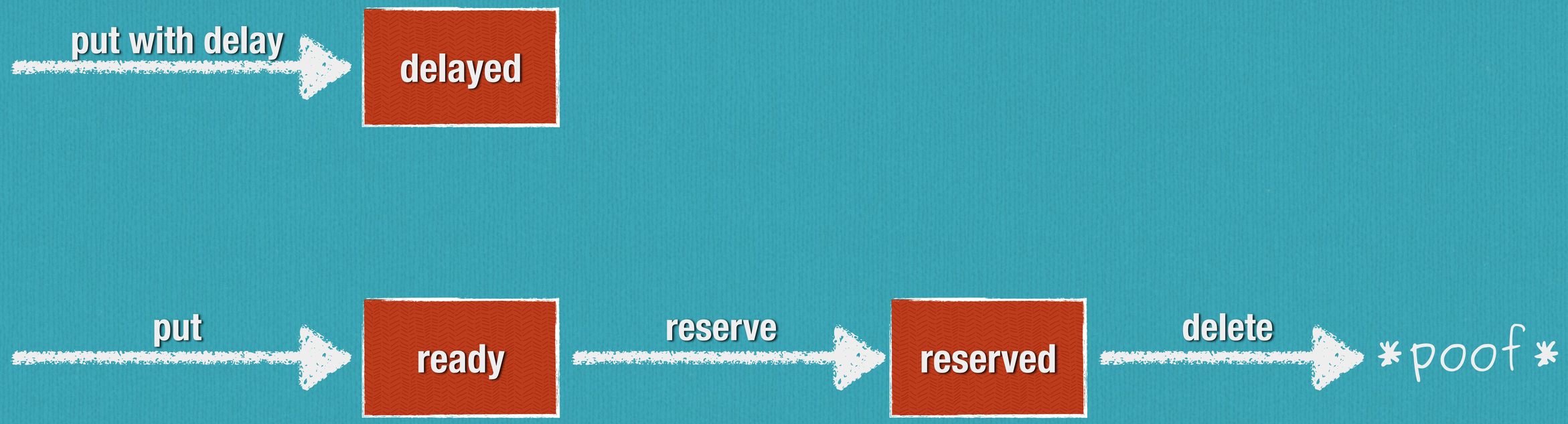
(mostly general) implementation best practices

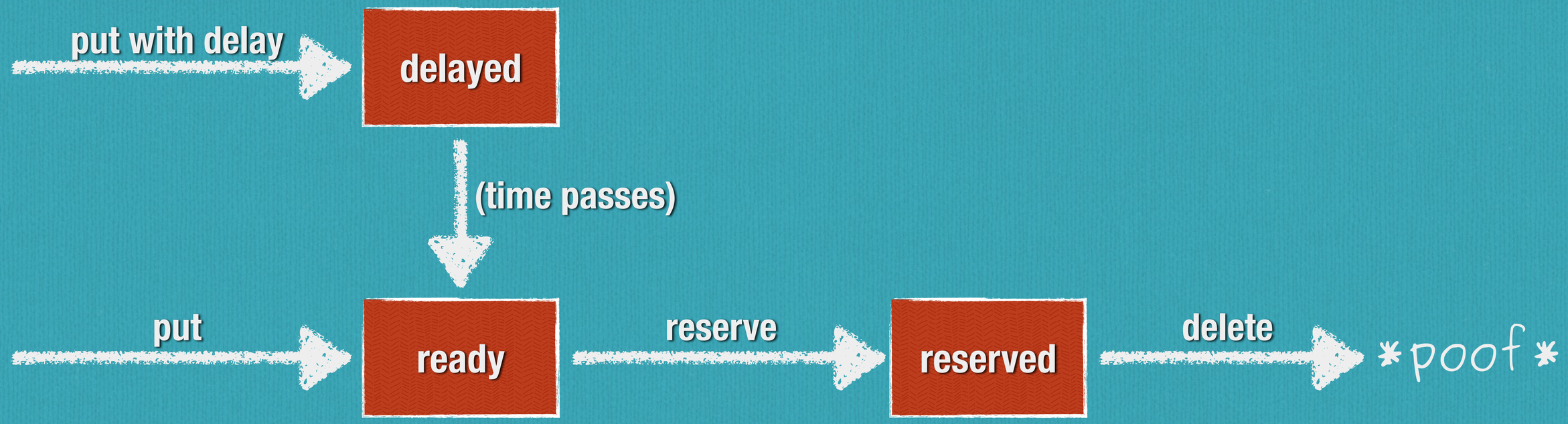

put
ready

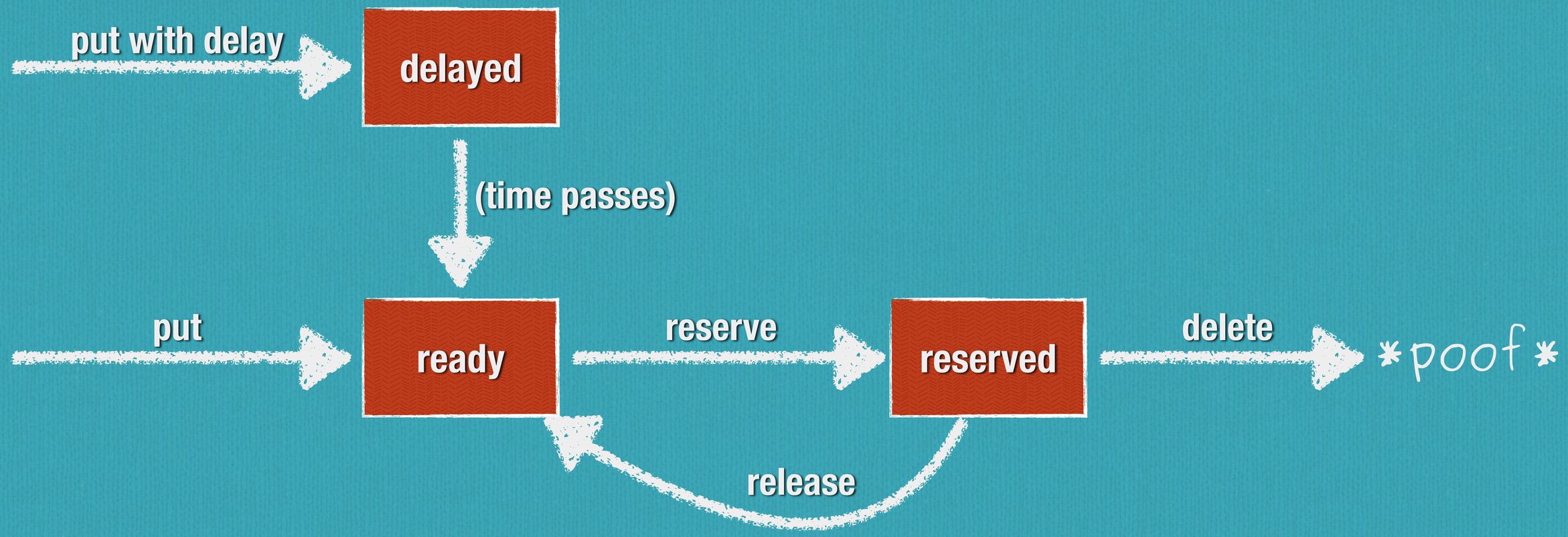
put
ready
reserve
reserved

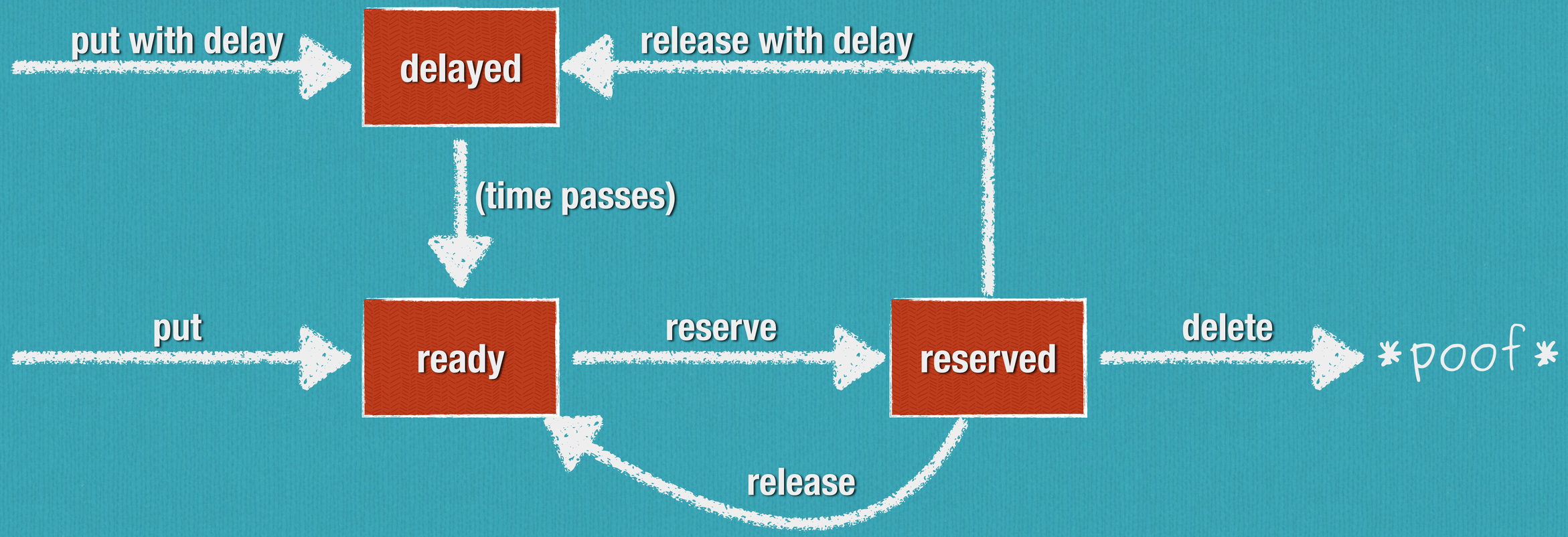
put
ready
reserve
reserved
delete
poof

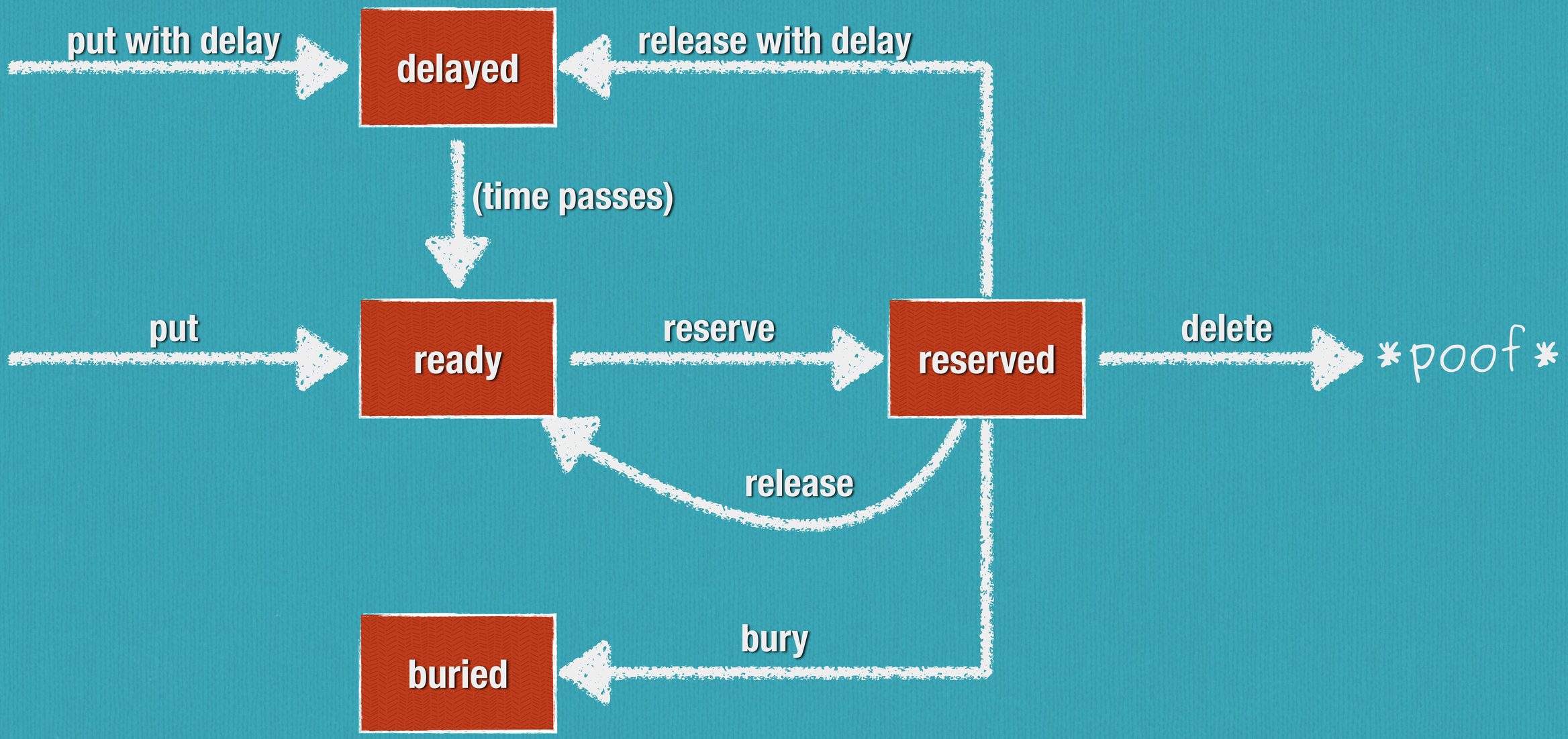
put
ready
reserve
reserved
delete
poof

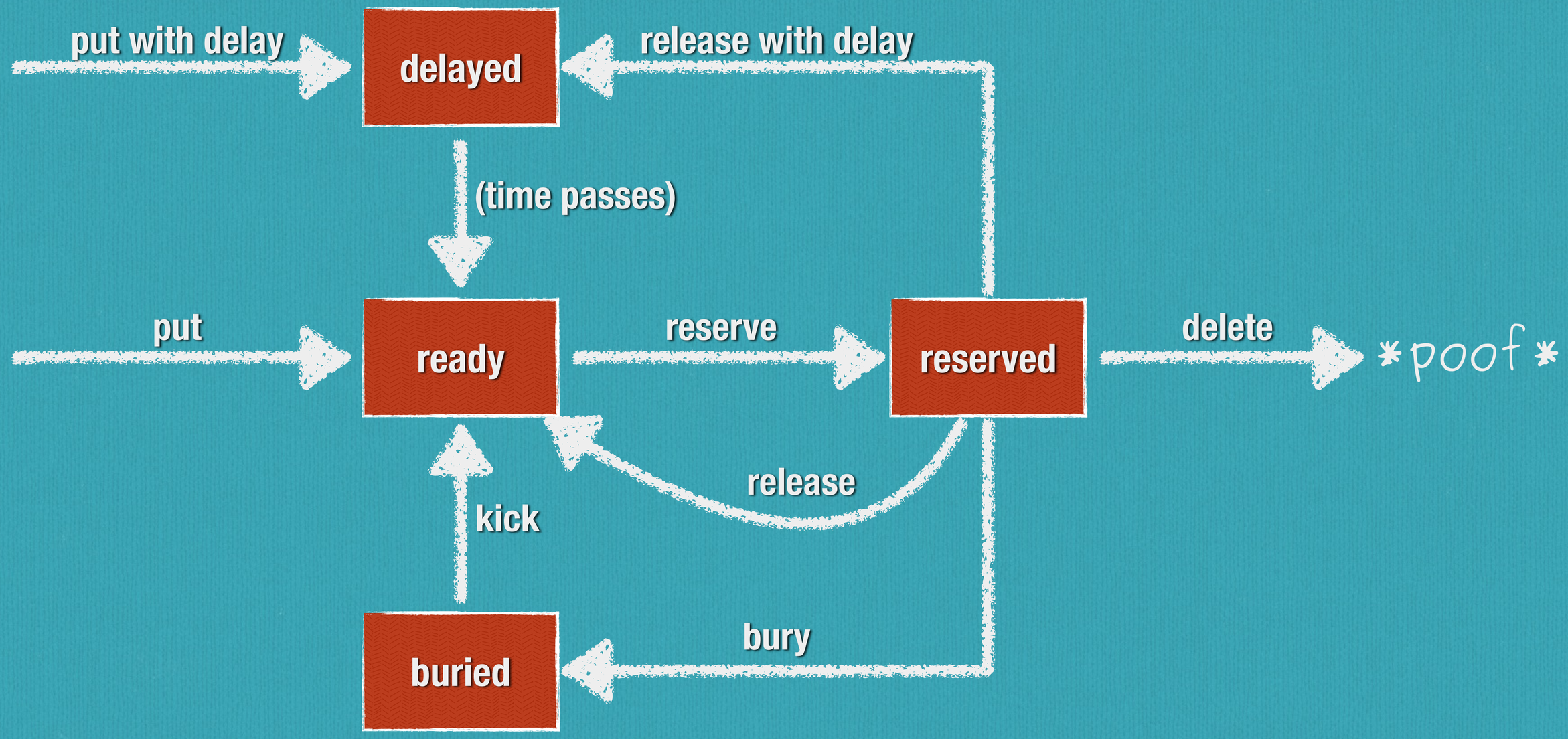

put with delay
delayed
put
ready
reserve
reserved
delete
poof

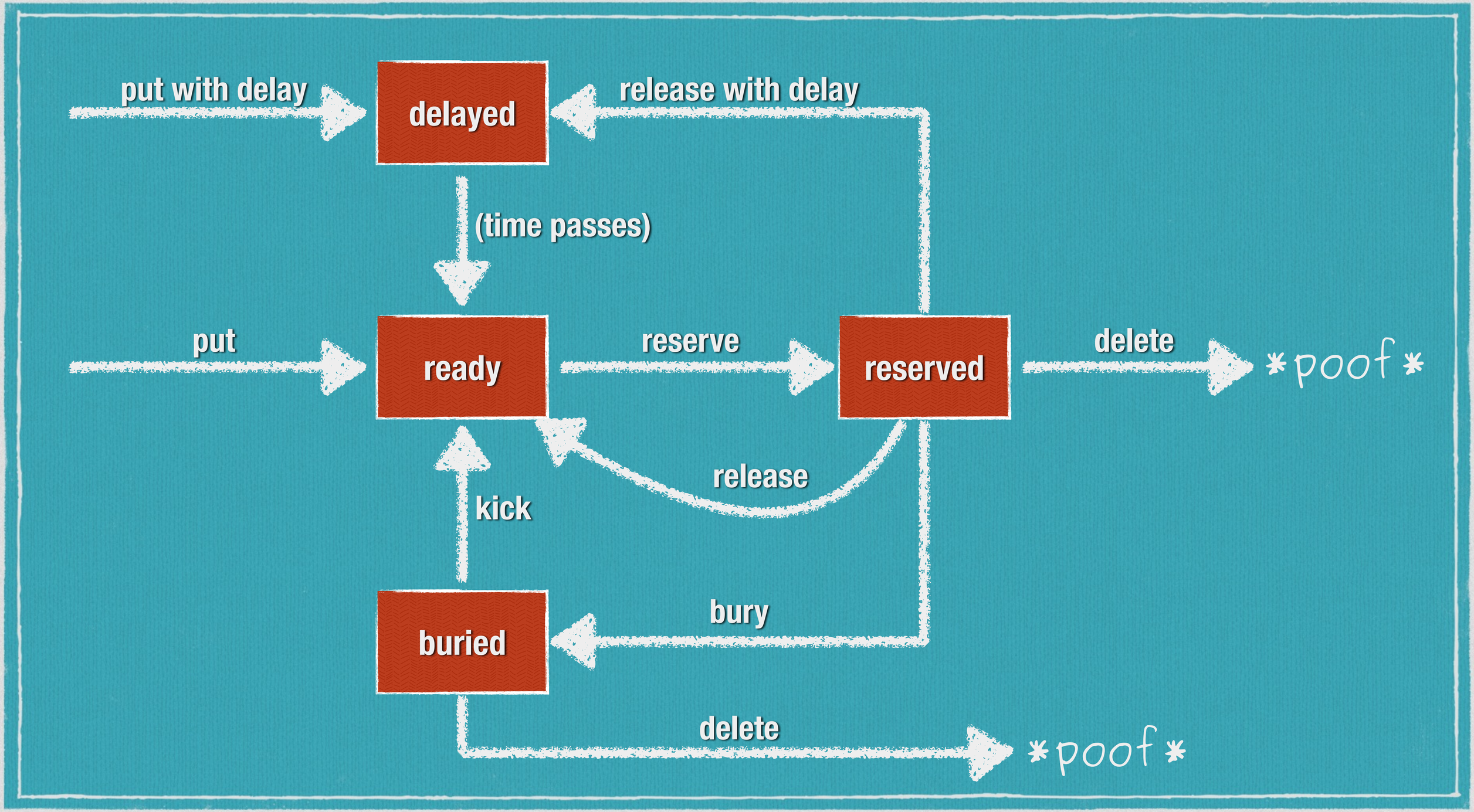

put with delay
delayed
(time passes)
put
ready
reserve
reserved
delete
poof


put with delay
delayed
(time passes)
put
ready
reserve
reserved
delete
poof
release


put with delay
delayed
release with delay
(time passes)
put
ready
reserve
reserved
delete
poof
release

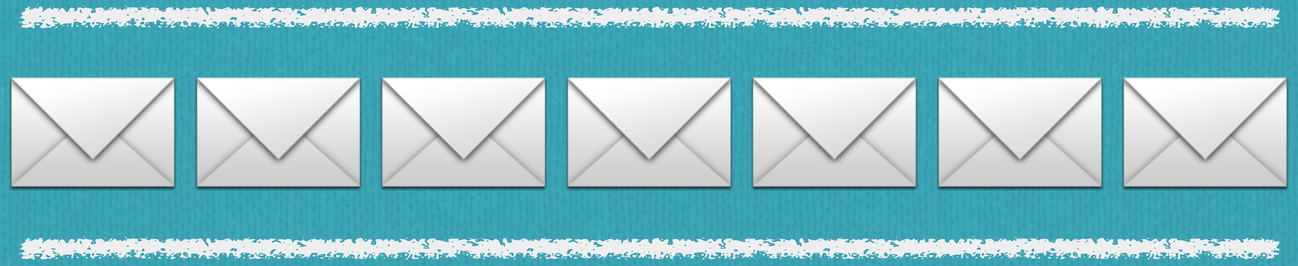

put with delay
delayed
release with delay
(time passes)
put
ready
reserve
reserved
delete
poof
release
bury
buried

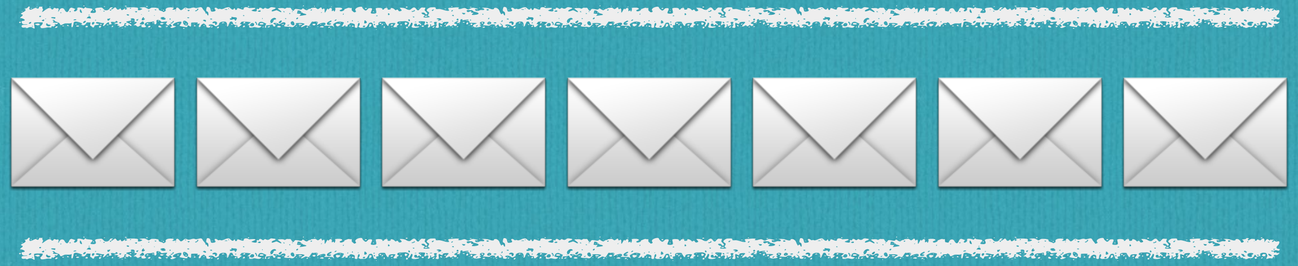

put with delay
delayed
release with delay
(time passes)
put
ready
reserve
reserved
delete
poof
release
kick
bury
buried


put with delay
delayed
release with delay
(time passes)
put
ready
reserve
reserved
delete
poof
release
kick
bury
buried
delete
poof

jobs will fail

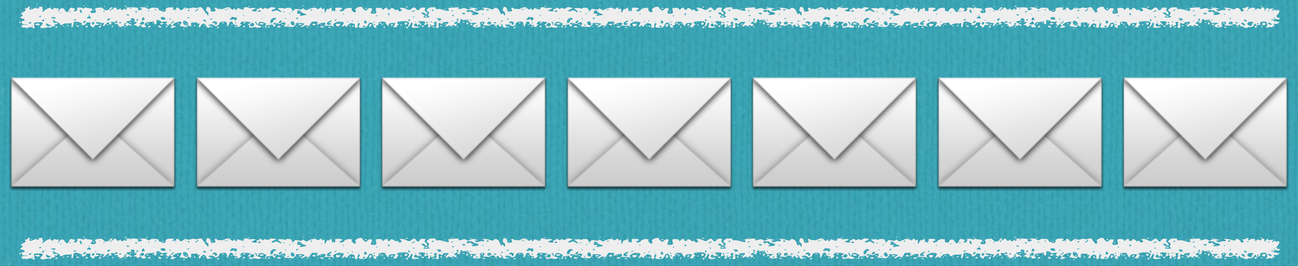
you should retry





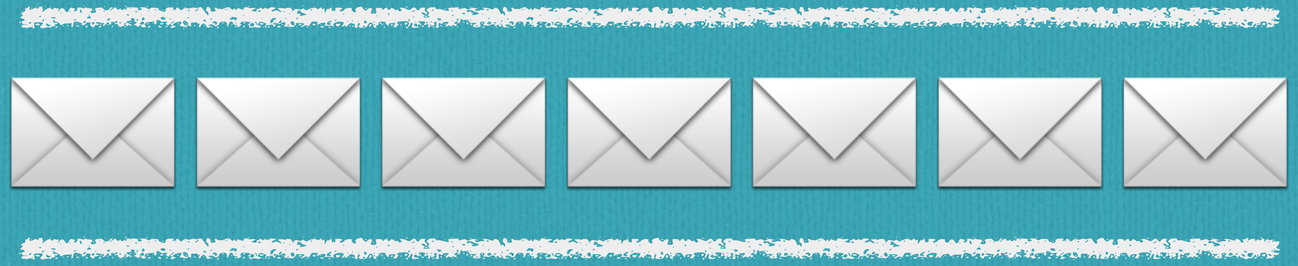
jobs will fail often

use timing for retries



don’t retry infinite

count your retries

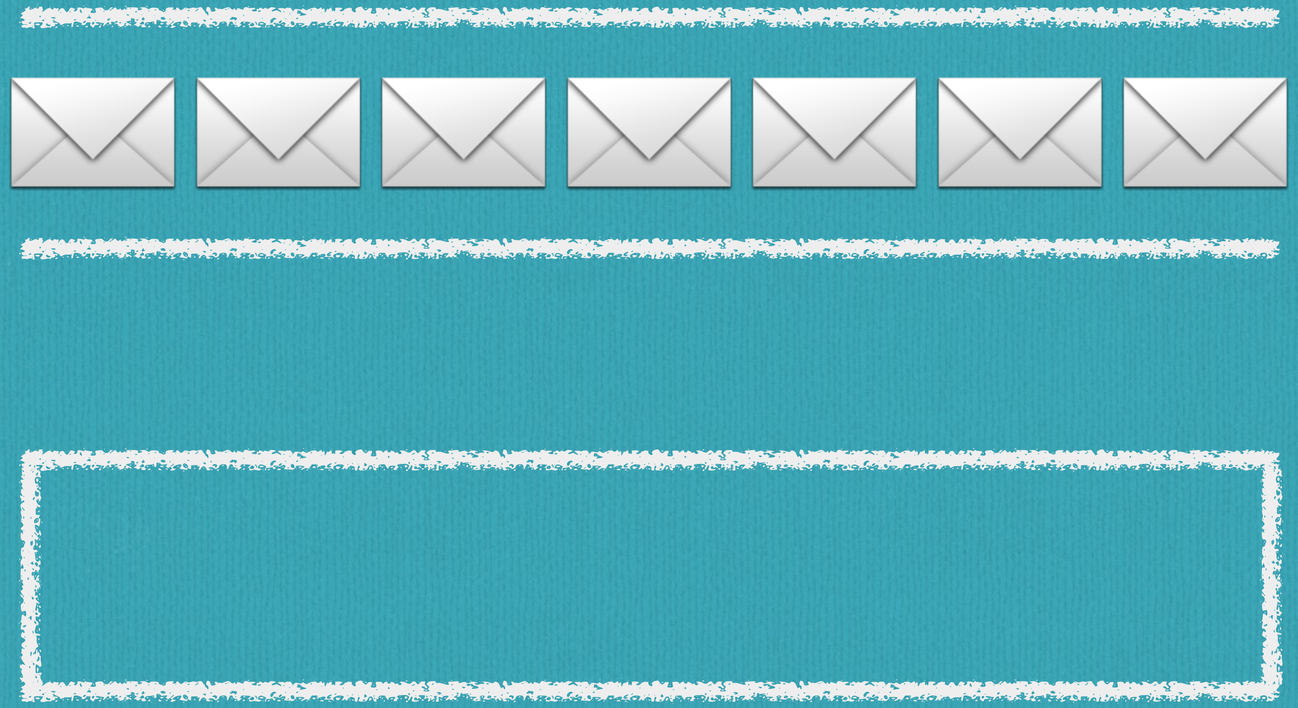


3

jobs can’t be done

separate them

99

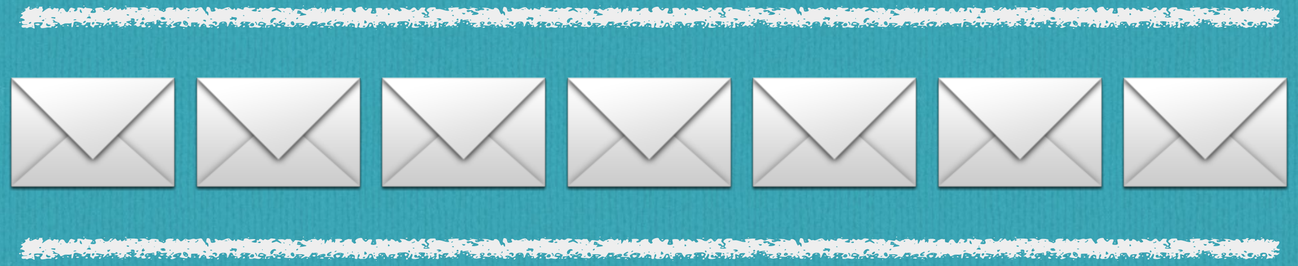


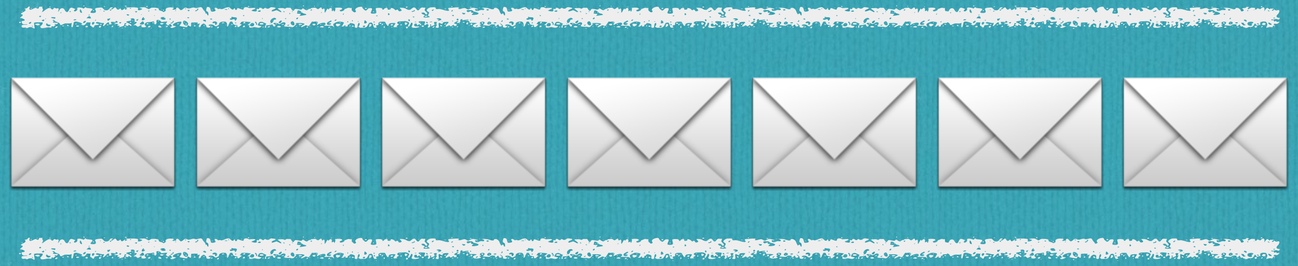
100

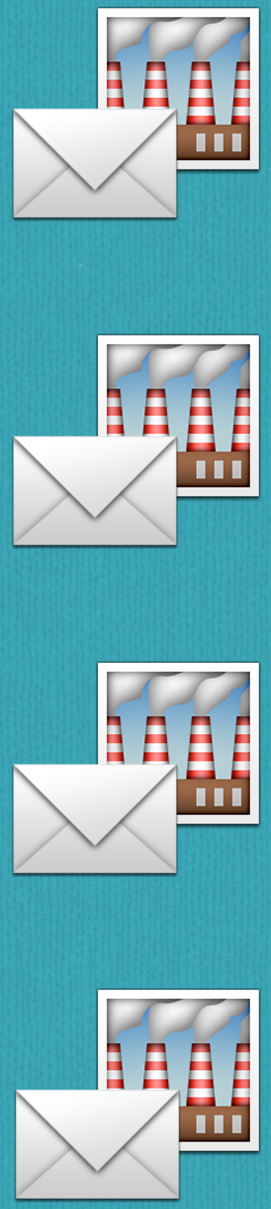
DEAD LETTER QUEUE

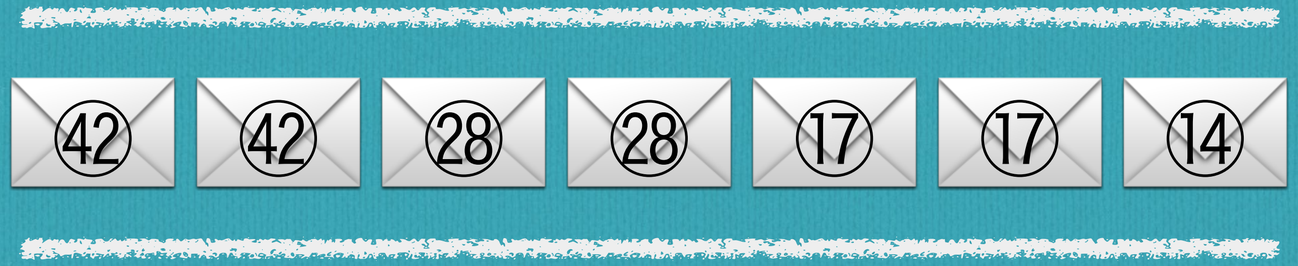
fanout & prioritize

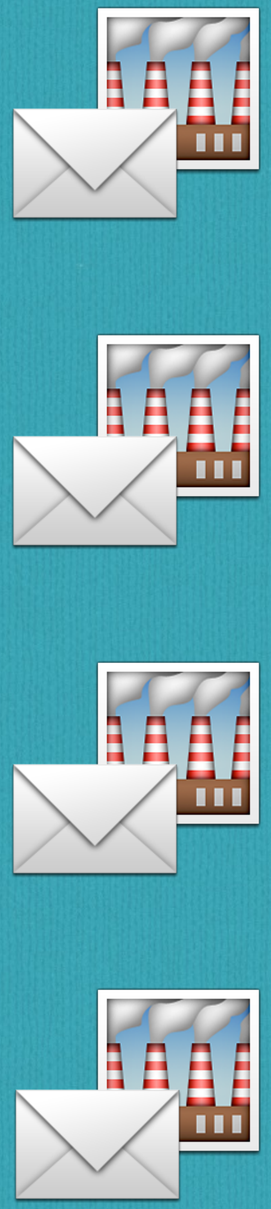
scale workers effectively








42
42
28
28
17
17
14



watch your queue

monitor and measure what is going on

Sidekiq Pro

Dashboard | Workers | Queues | Retries | Scheduled | Batches

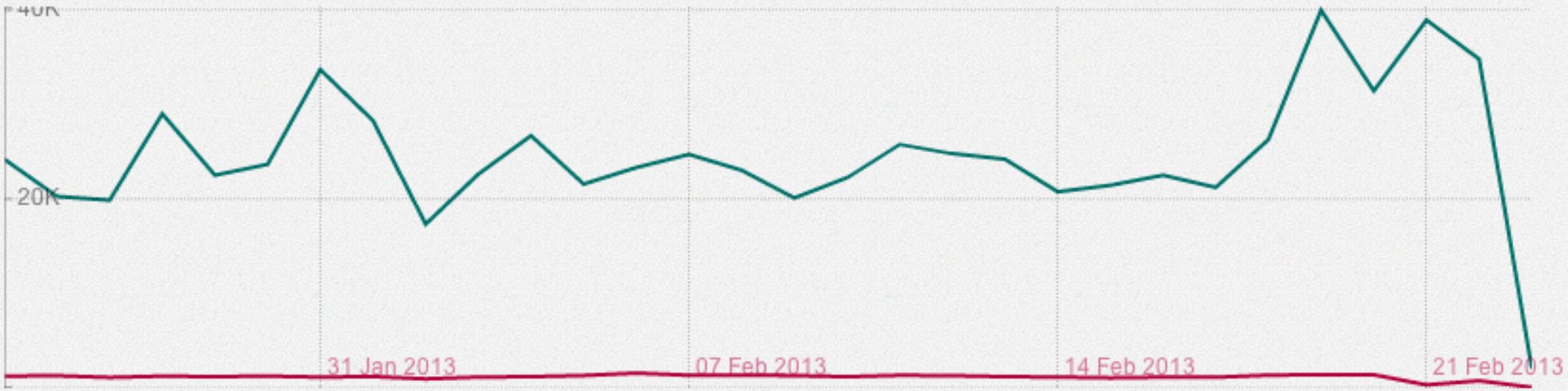
Status idle

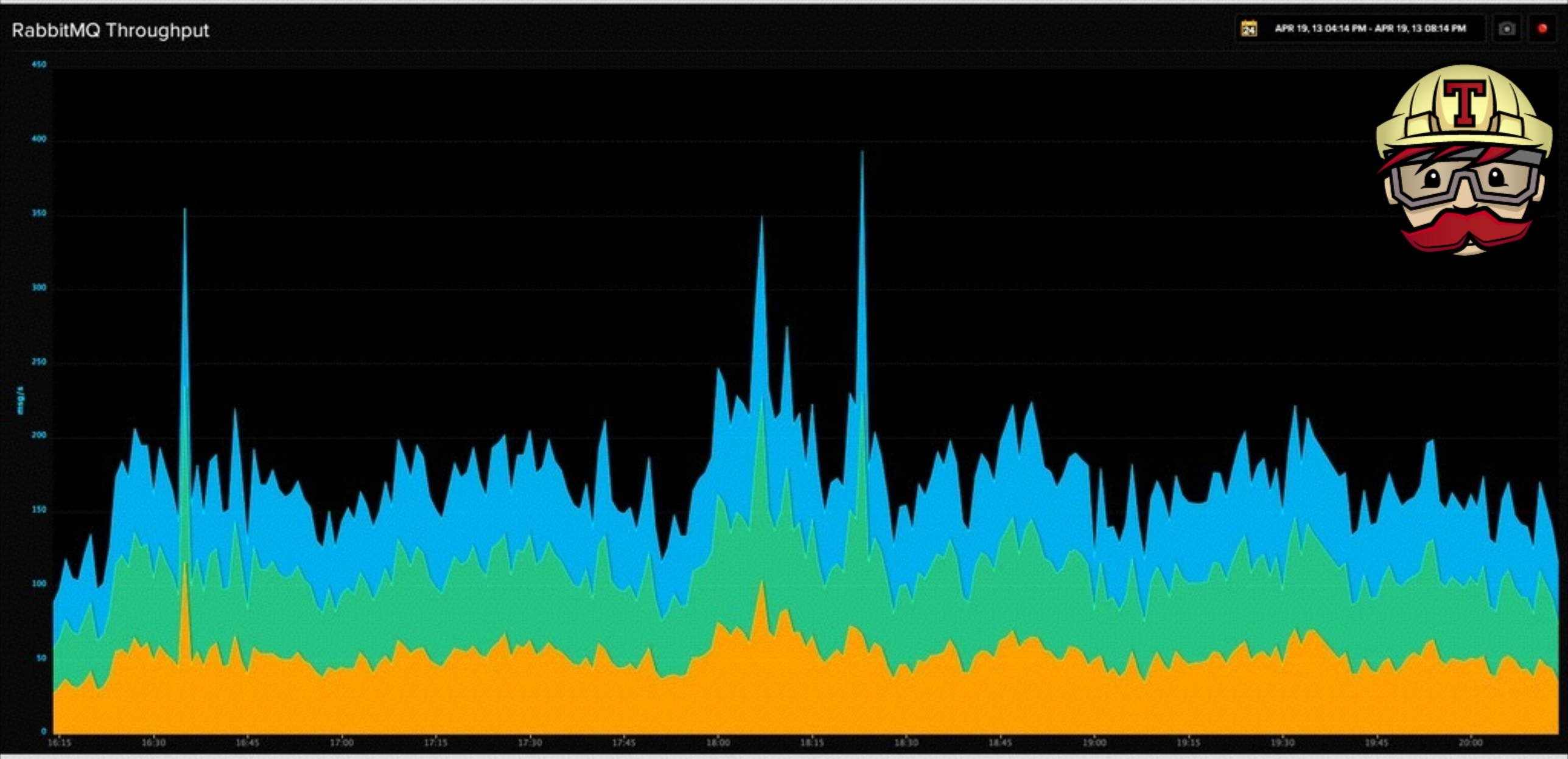
Processed	4,846,522
Failed	170,661
Busy	1
Scheduled	12
Retries	433
Enqueued	0

Dashboard

Real-time

History 1 week 1 month 3 month 6 month




RabbitMQ Throughput
APR 19, 13:04:14 PM - APR 19, 13:08:14 PM
msg/s
450
400
350
300
250
200
150
100
50
0
16:15
16:30
16:45
17:00
17:15
17:30
17:45
18:00
18:15
18:30
18:45
19:00
19:15
19:30
19:45
20:00

Yay or Nay?

Thank You

Let’s Talk

or just ping me later @codestars or ole@jimdo.com

continuous reading

- http://blog.thecodepath.com/2013/01/06/asynchronous-processing-in-web-applications-part-2-developers-need-to-understand-message-queues/
- Mathias Meyer - Message Queues - Railsberry 2013 https://vimeo.com/68698333
- https://github.com/kr/beanstalkd
- https://github.com/dp-opensource/pheanstalk
- https://en.wikipedia.org/wiki/Message_queue
- https://github.com/denniskuczynski/beanstalkd_view
- https://github.com/twitter/kestrel
- https://github.com/kr/beanstalkd/blob/master/doc/protocol.txt

I want to thank my company Jimdo, my girlfriend and all my coworkers for supporting me!

Products / Name dropping

- activemq
- rabbitmq
- Ømq (zeromq)
- zookeeper
- pubnub
- ironmq
- sqs
- sidekiq
- resque
- kue
- gearman
- beanstalkd
- kestrel
- kafka
- delayed_job
- celery
- qless
- mqseries
- nsq
- jms
- qpid
- hornetq
- swiftmq
- mqseries
- stormmq
- dotnetmq
- snakemq
- dropr

AND ALWAYS REMEMBER:

SAYS