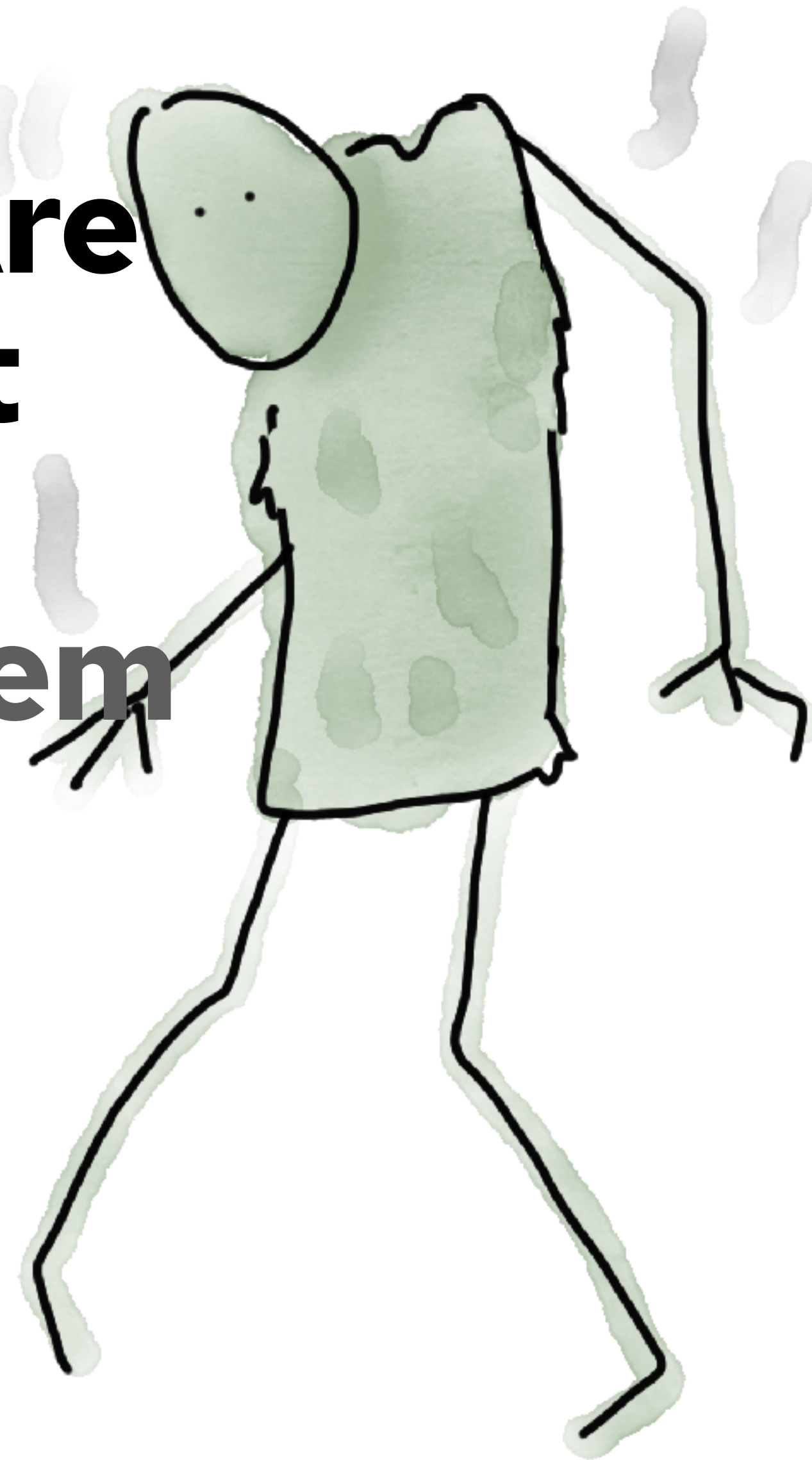


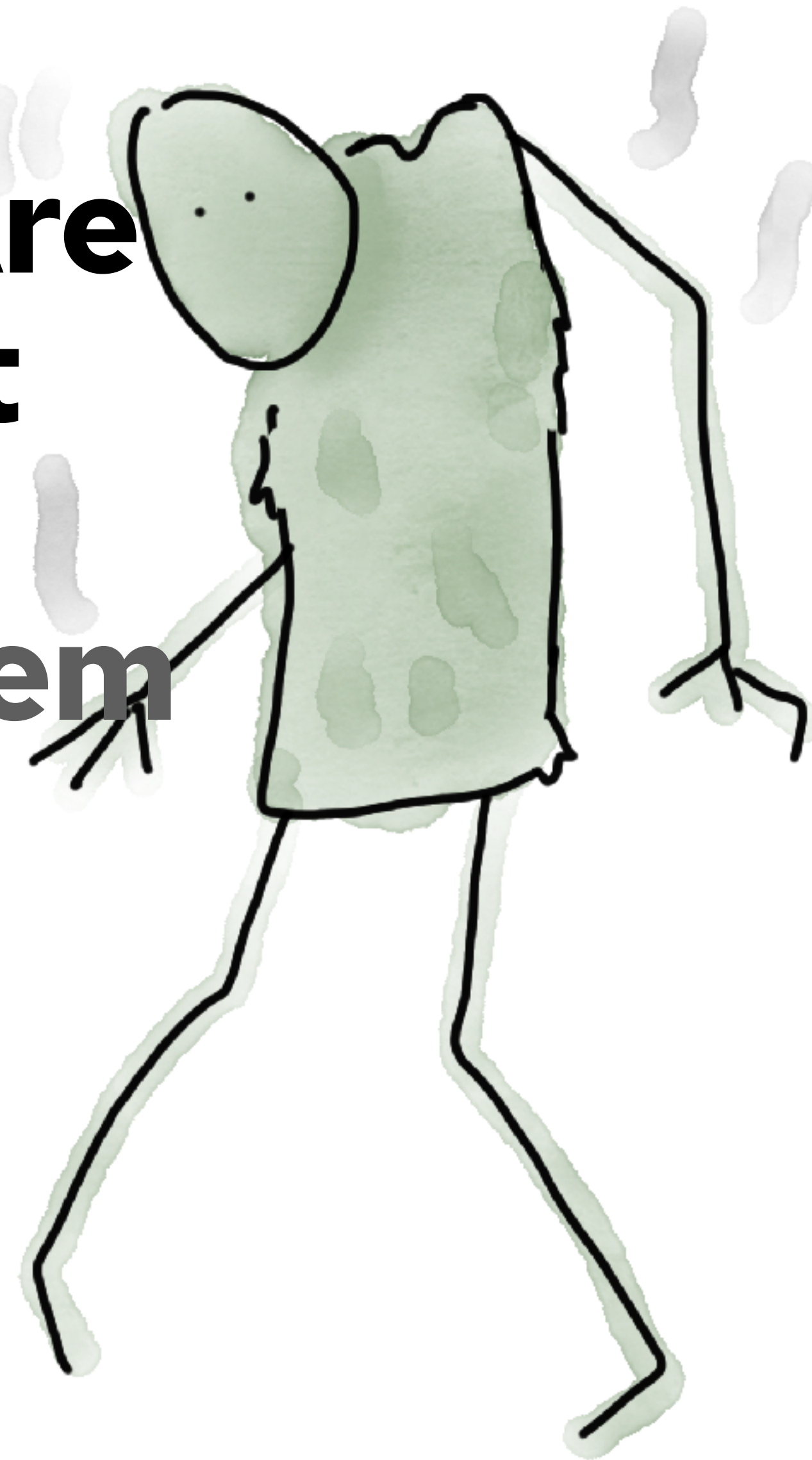
# Why Cloud Zombies Are Destroying the Planet and How You Can Stop Them



**Holly Cummins**  
**Red Hat**

QCon London | March 29, 2023

# Why Cloud Zombies Are Destroying the Planet and How You Can Stop Them



**Holly Cummins**  
**Red Hat**

QCon London | March 29, 2023



**Wes Bos**  @wesbos · Jan 12, 2022



Replying to [@wesbos](#)

I get charged ~\$2 a month from AWS and I'm too scared to turn it off and too lazy to figure out what is causing the bandwidth

I still get emails from a ~8 year old Client WordPress install, that I'm pretty sure is an on-prem server. No idea how to access it, but it emails me



11



3



93





**Wes Bos**  @wesbos · Jan 12, 2022

I just turned off a digital ocean droplet that I created in 2013

I just deleted a snapshot from 2014 that I've been paying to 74 cents a month store for 7 years.

Death by 1000 cuts.



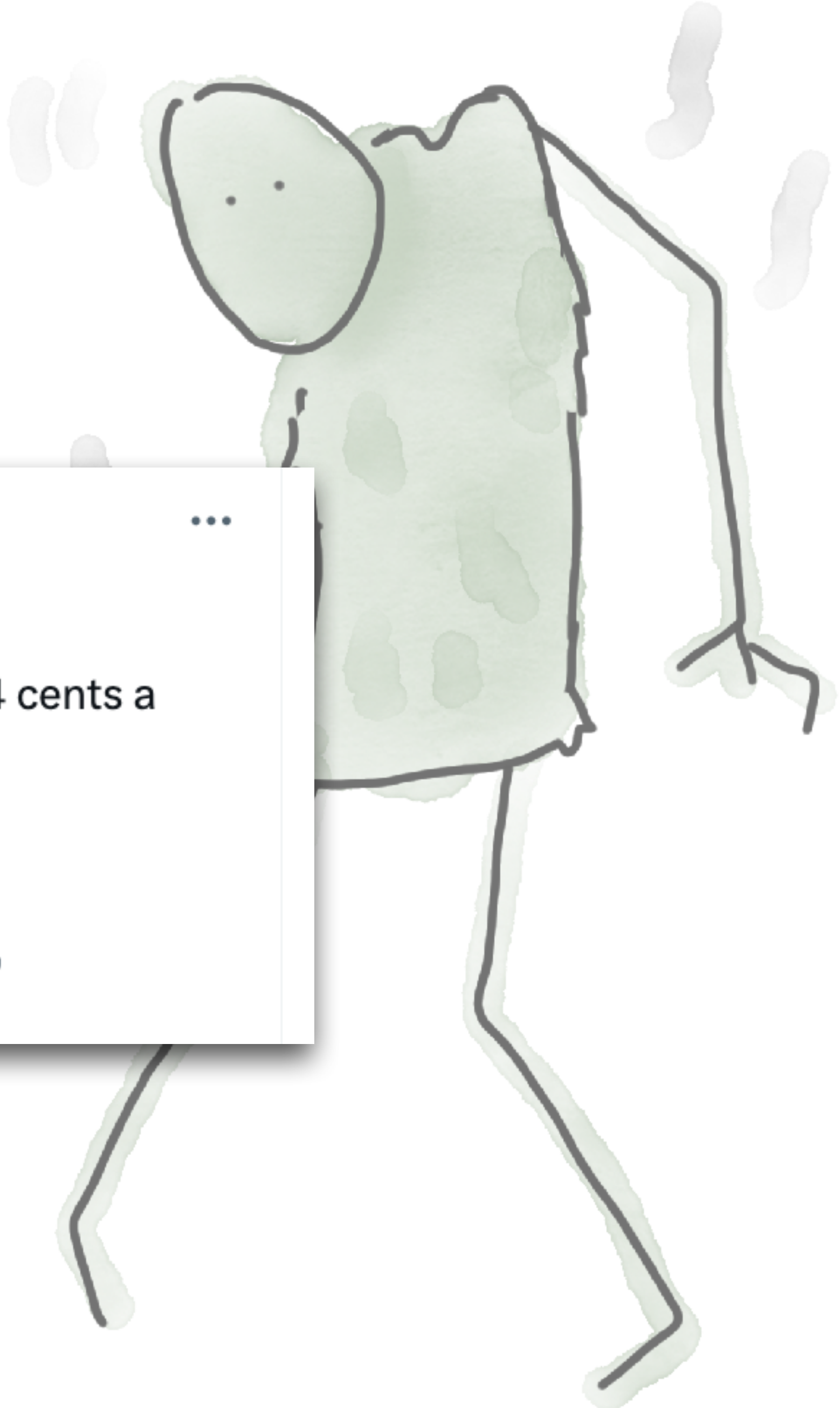
3



2



84



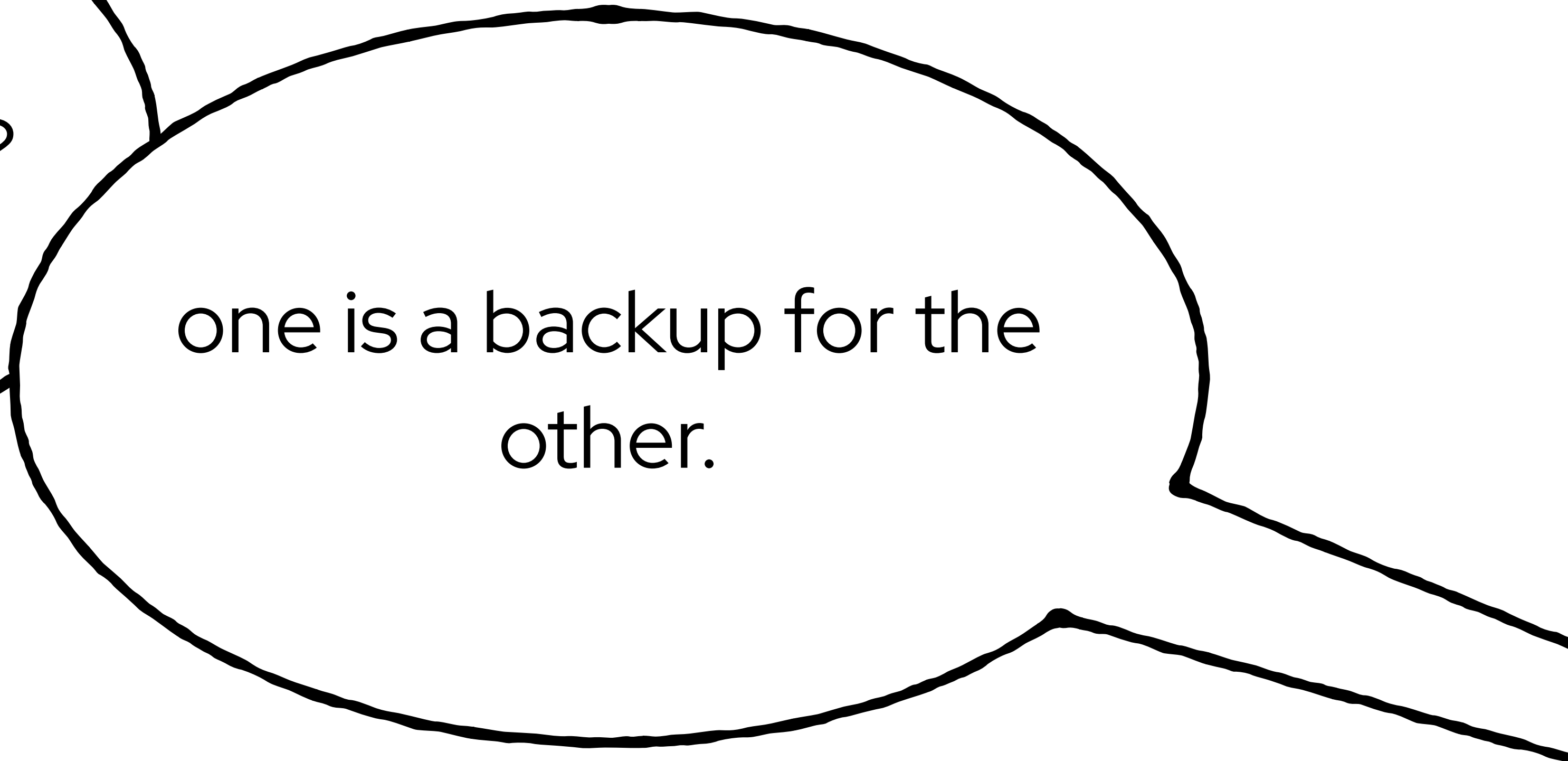




what do these servers do?



what do these servers do?



one is a backup for the  
other.

what do these servers do?

one is a backup for the  
other.

yes, but what do they do?



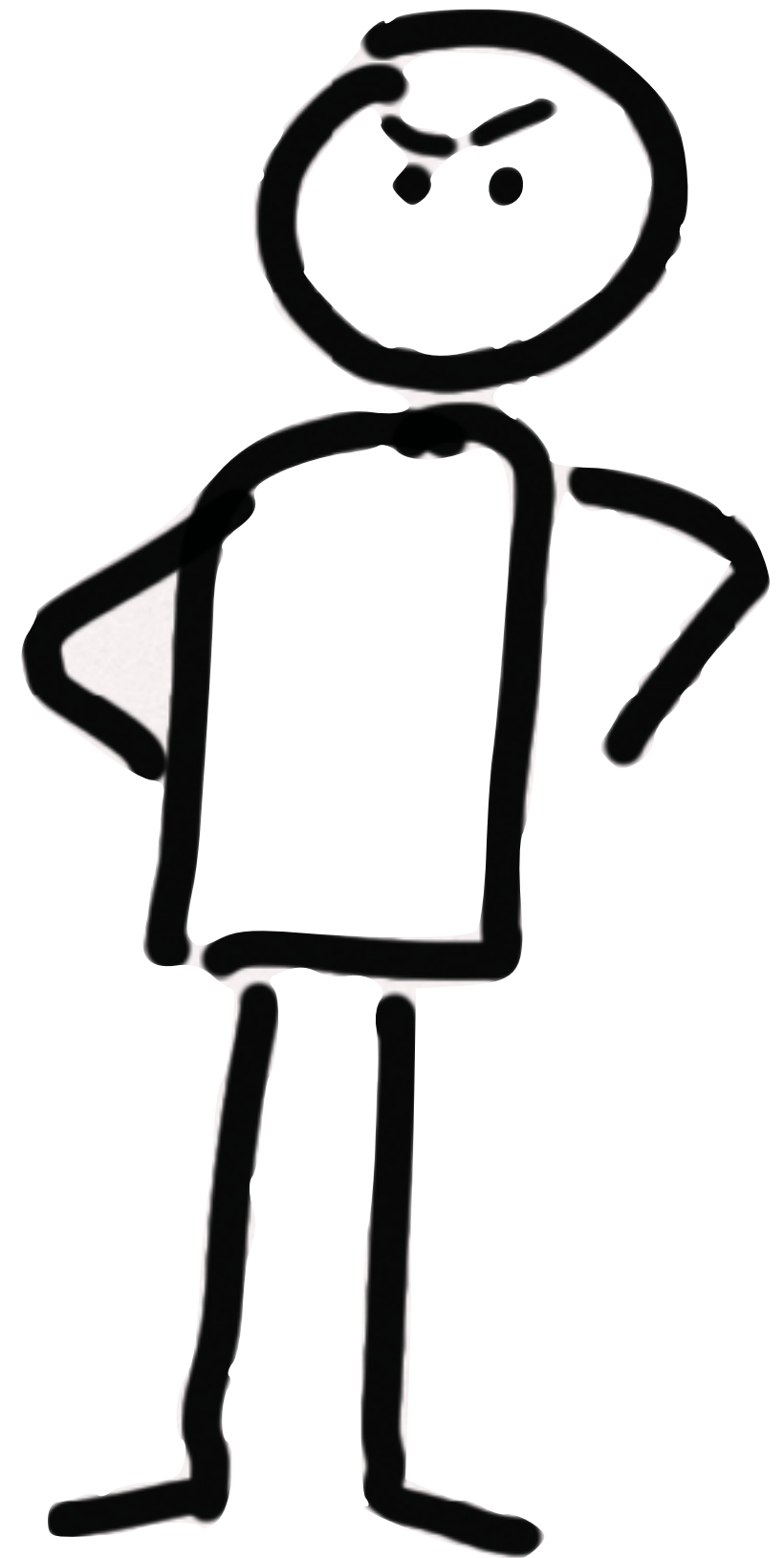
what do these servers do?

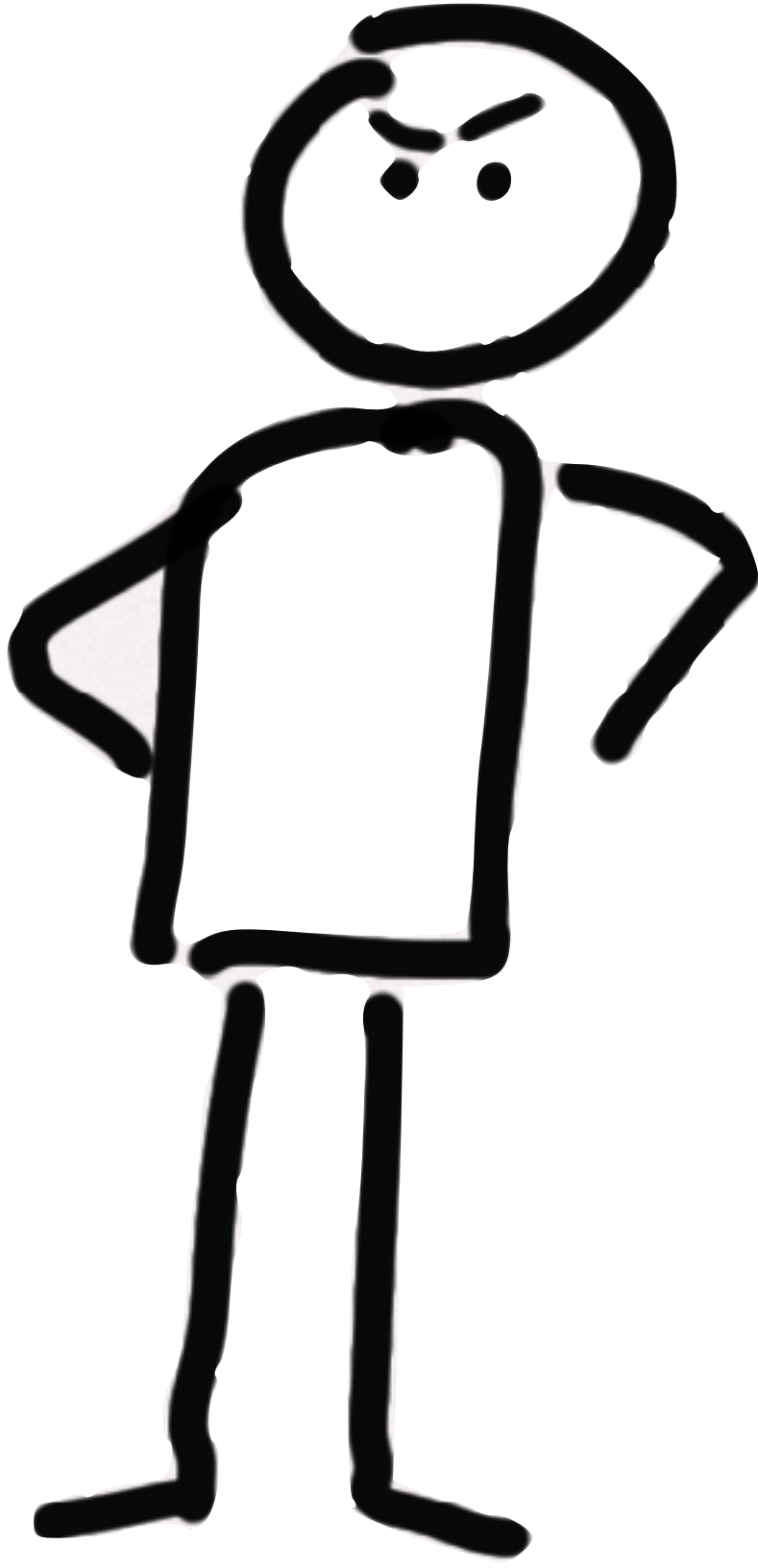
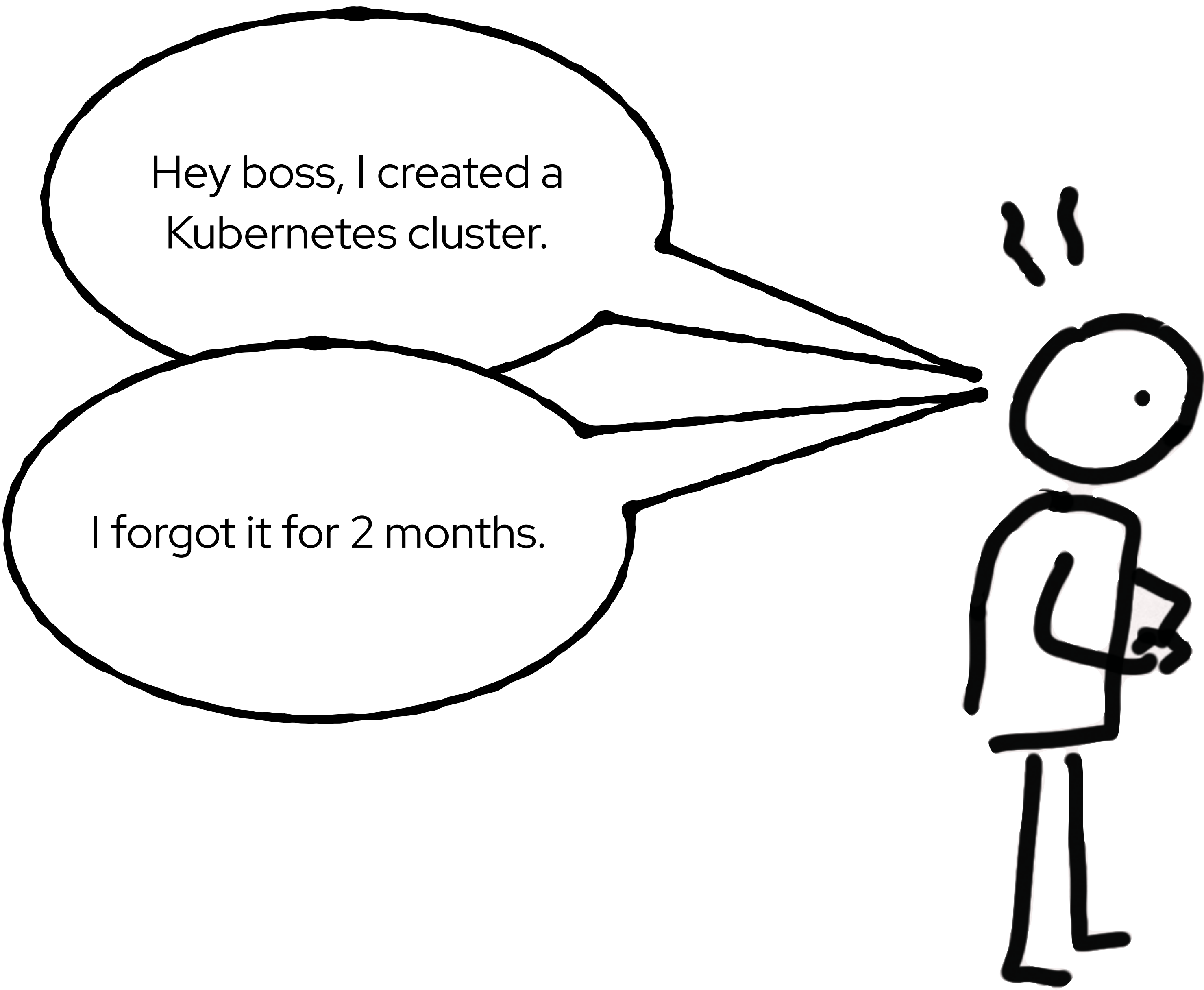
one is a backup for the other.

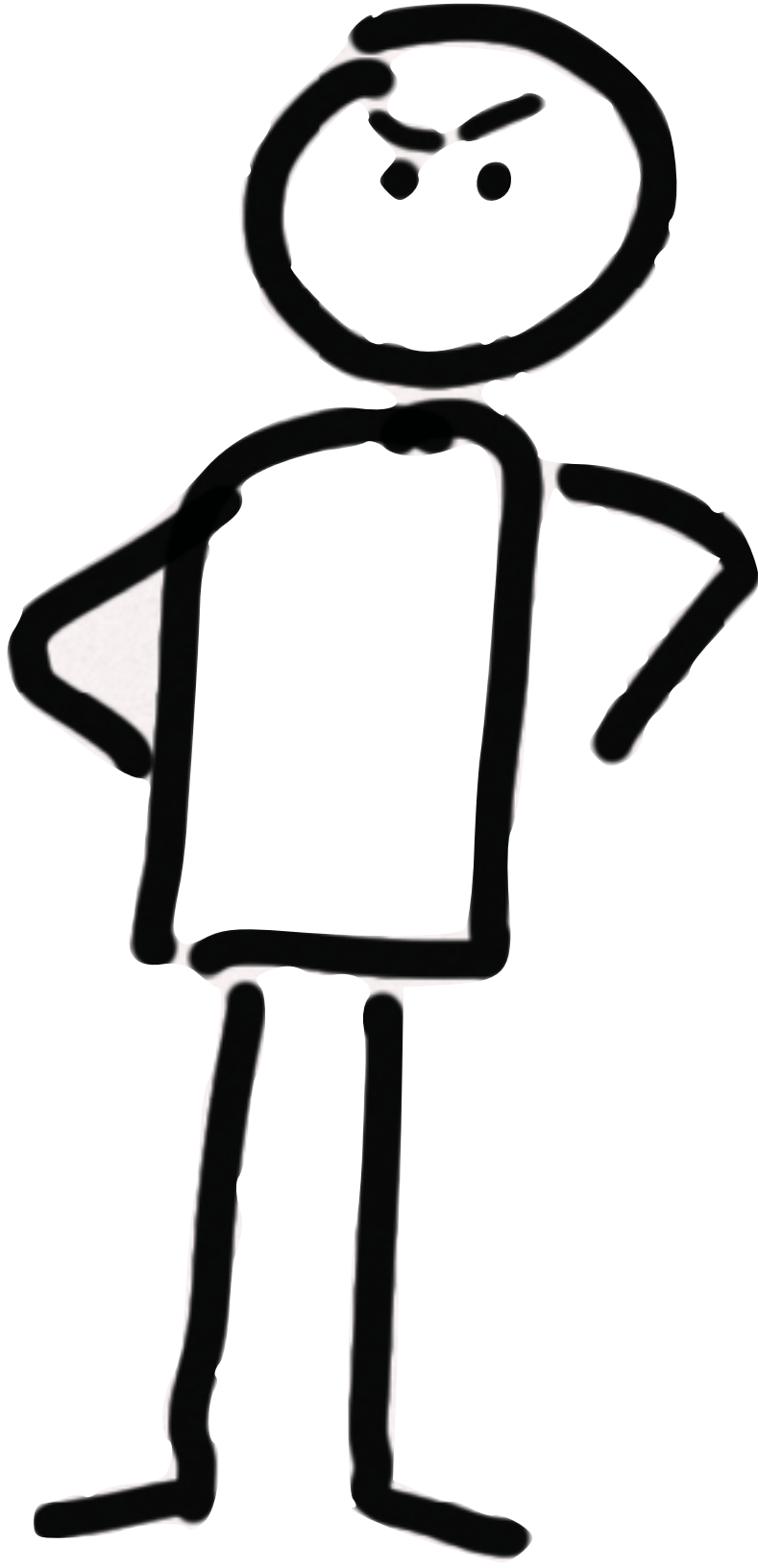
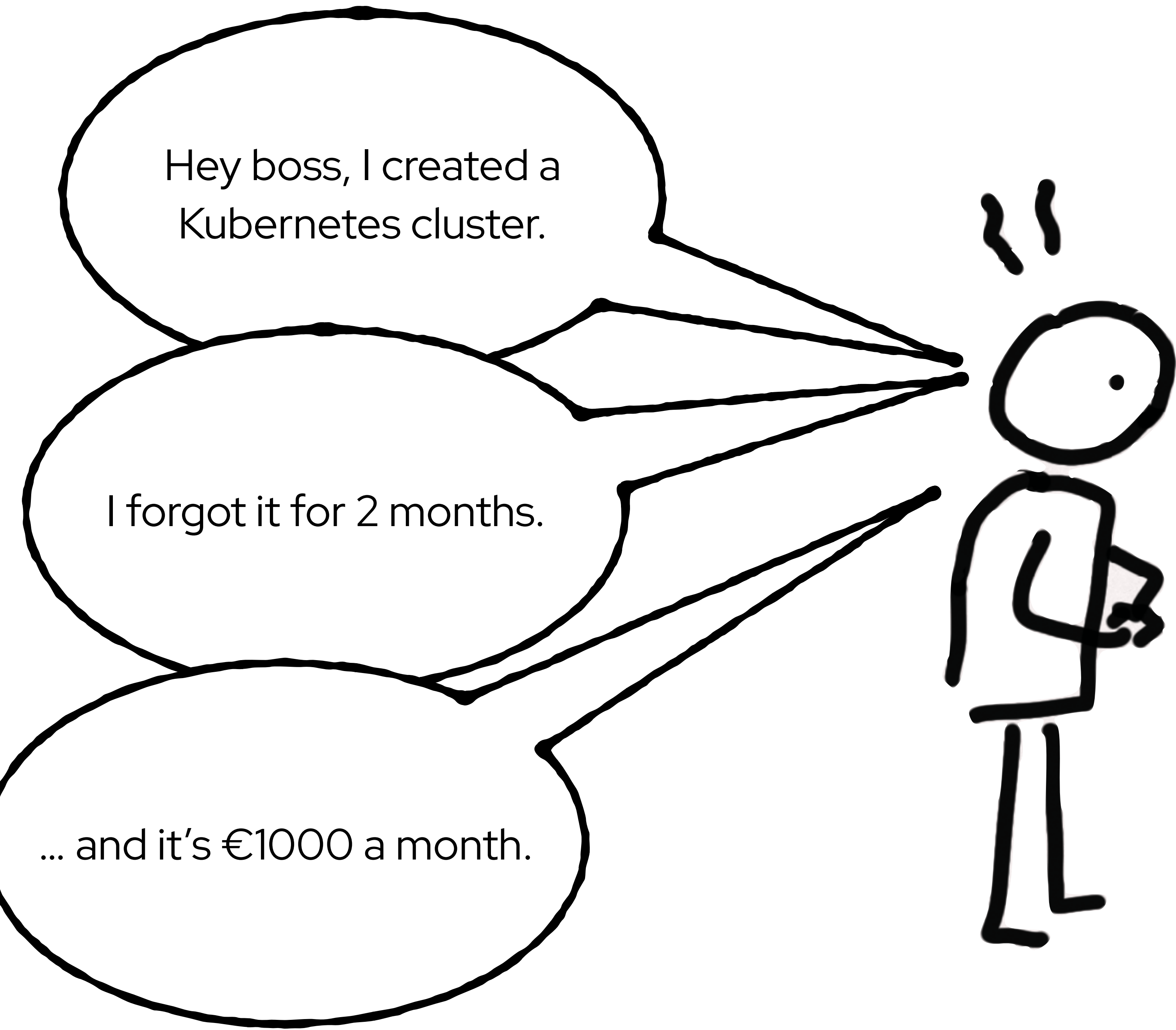
yes, but what do they do?

no one has known for a couple of decades

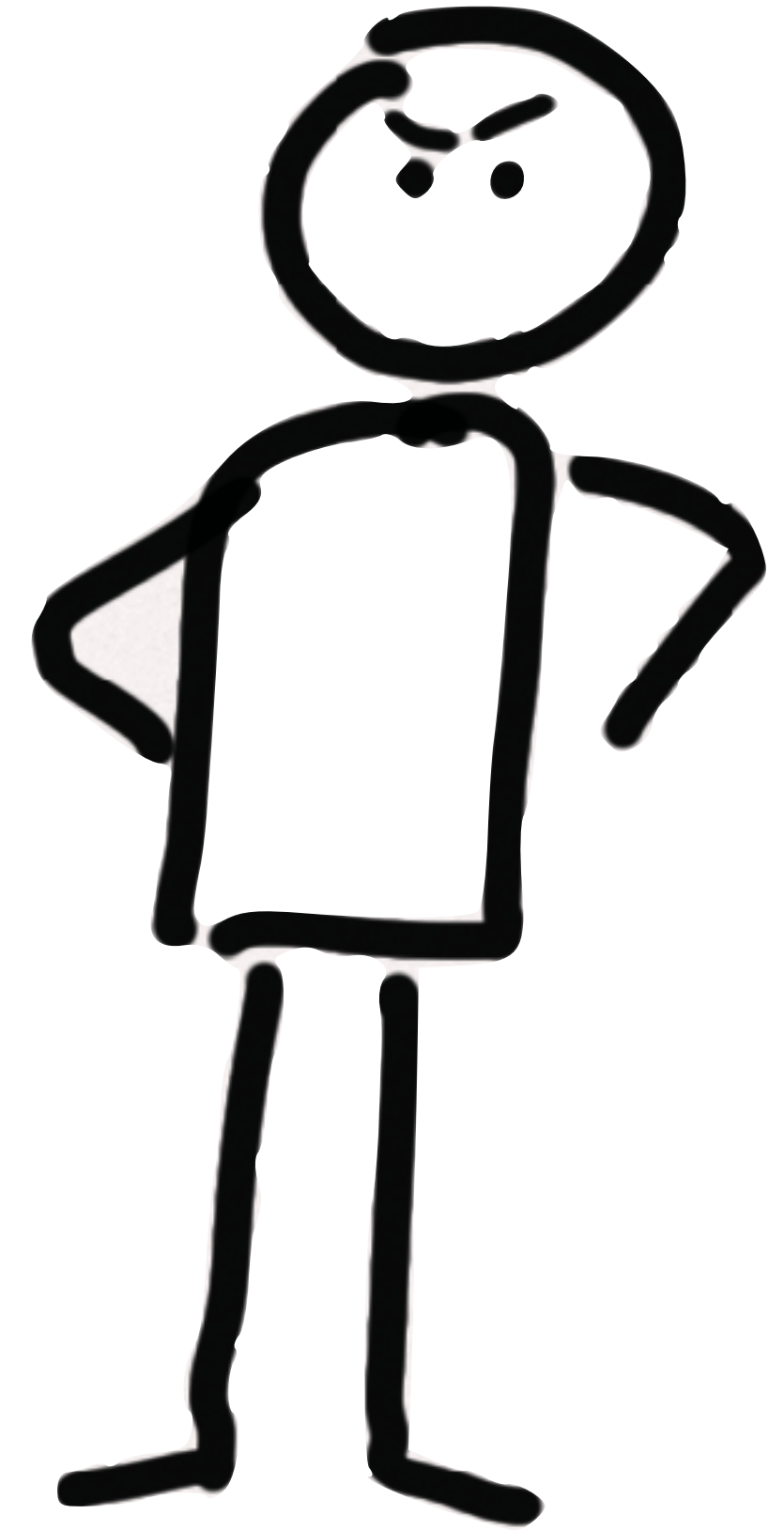
Hey boss, I created a  
Kubernetes cluster.





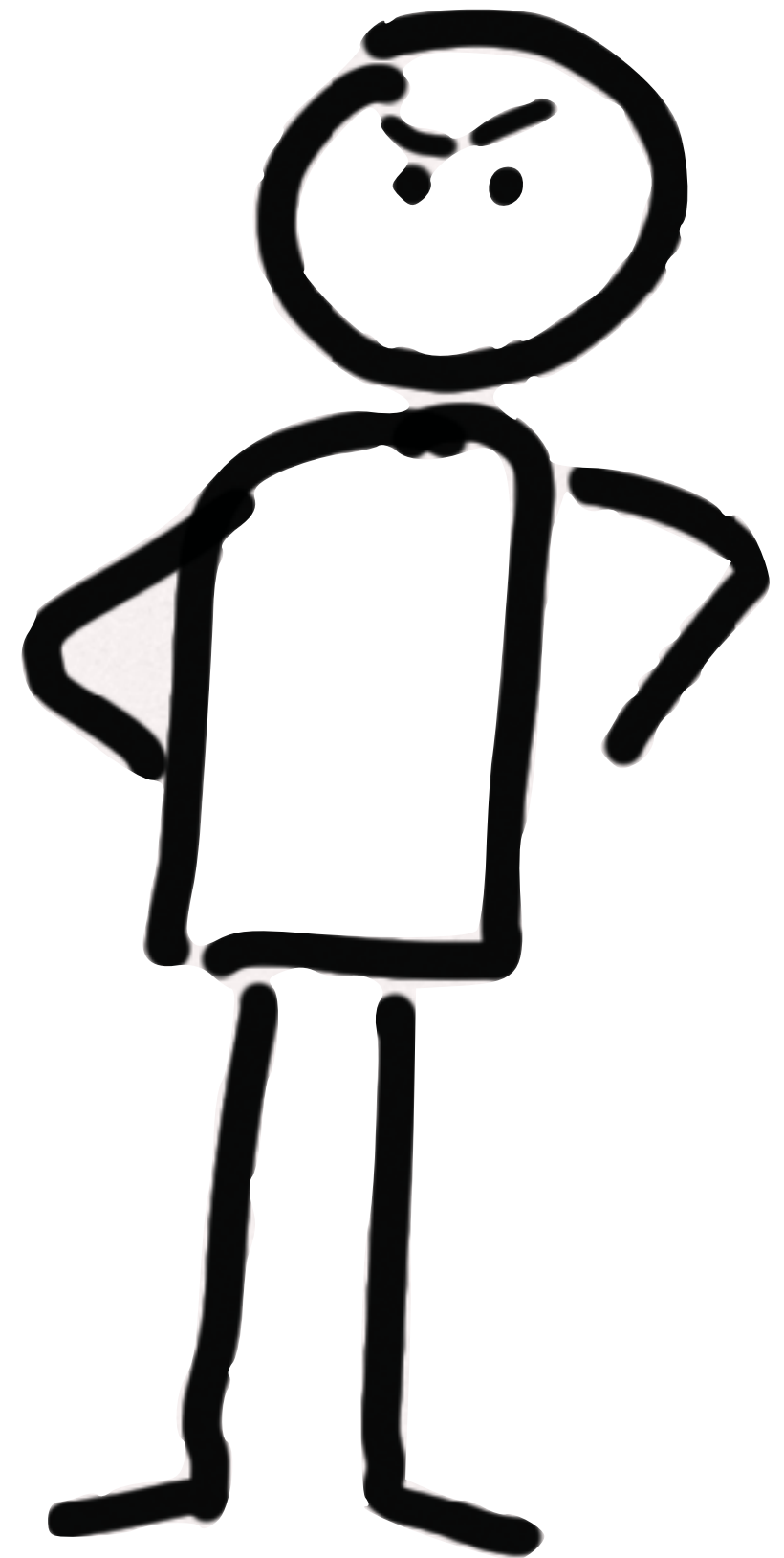


Hey boss, while I was working on a QCon talk about sustainability ...



Hey boss, while I was working on a QCon talk about sustainability ...

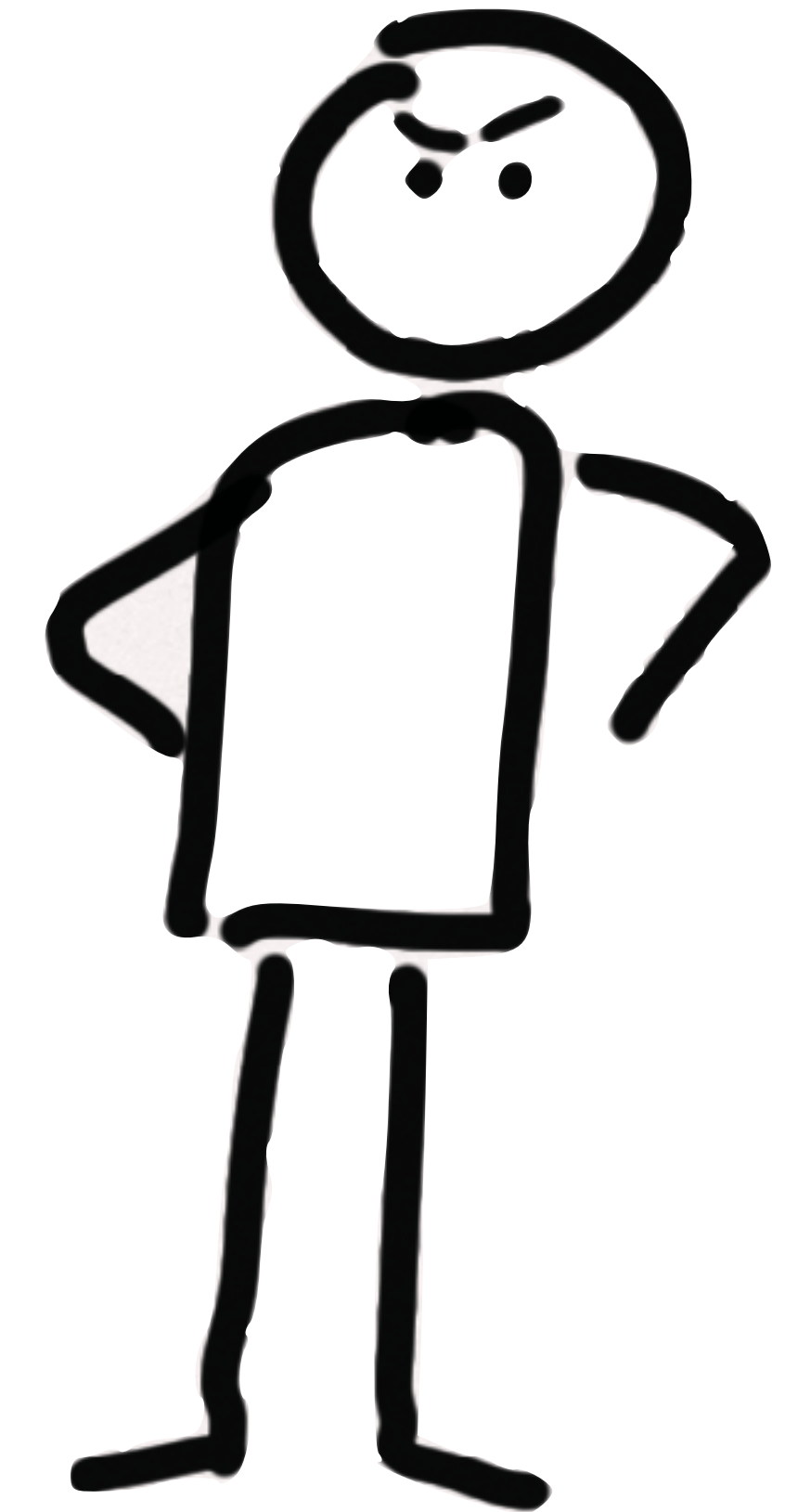
I left the Quarkus CI on Mac disabled

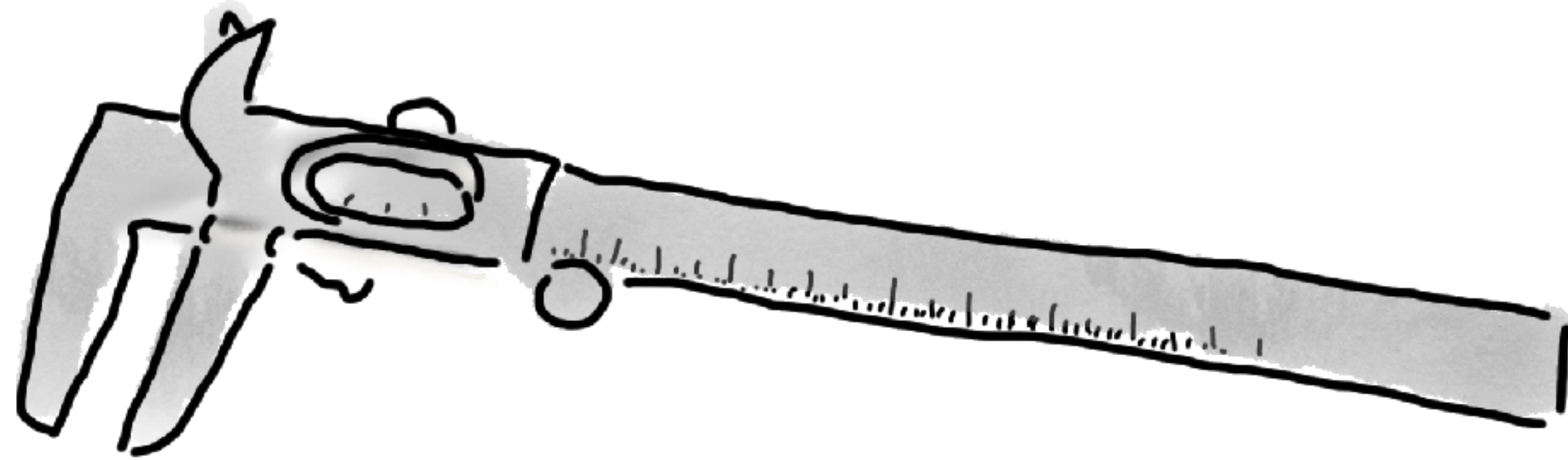


Hey boss, while I was working on a QCon talk about sustainability ...

I left the Quarkus CI on Mac disabled

... and the instance is \$159 a month.





“measure, don’t guess”

(or decide based on stories on the internet)



actual picture of a zombie  
(it's invisible)



actual picture of a zombie  
(it's invisible)

2015 survey

30%

of 4,000 servers doing  
**no** useful work



2017 survey

25%

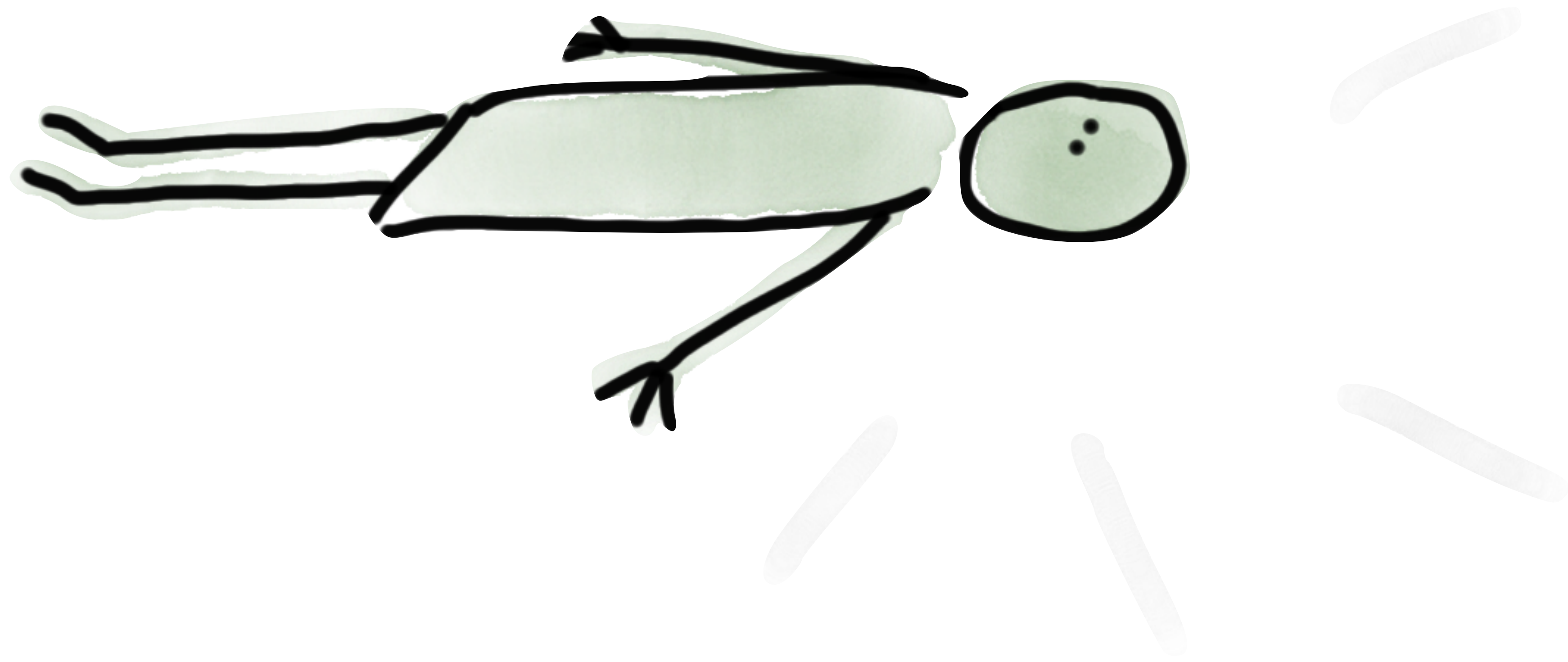
of 16,000 servers doing  
**no** useful work

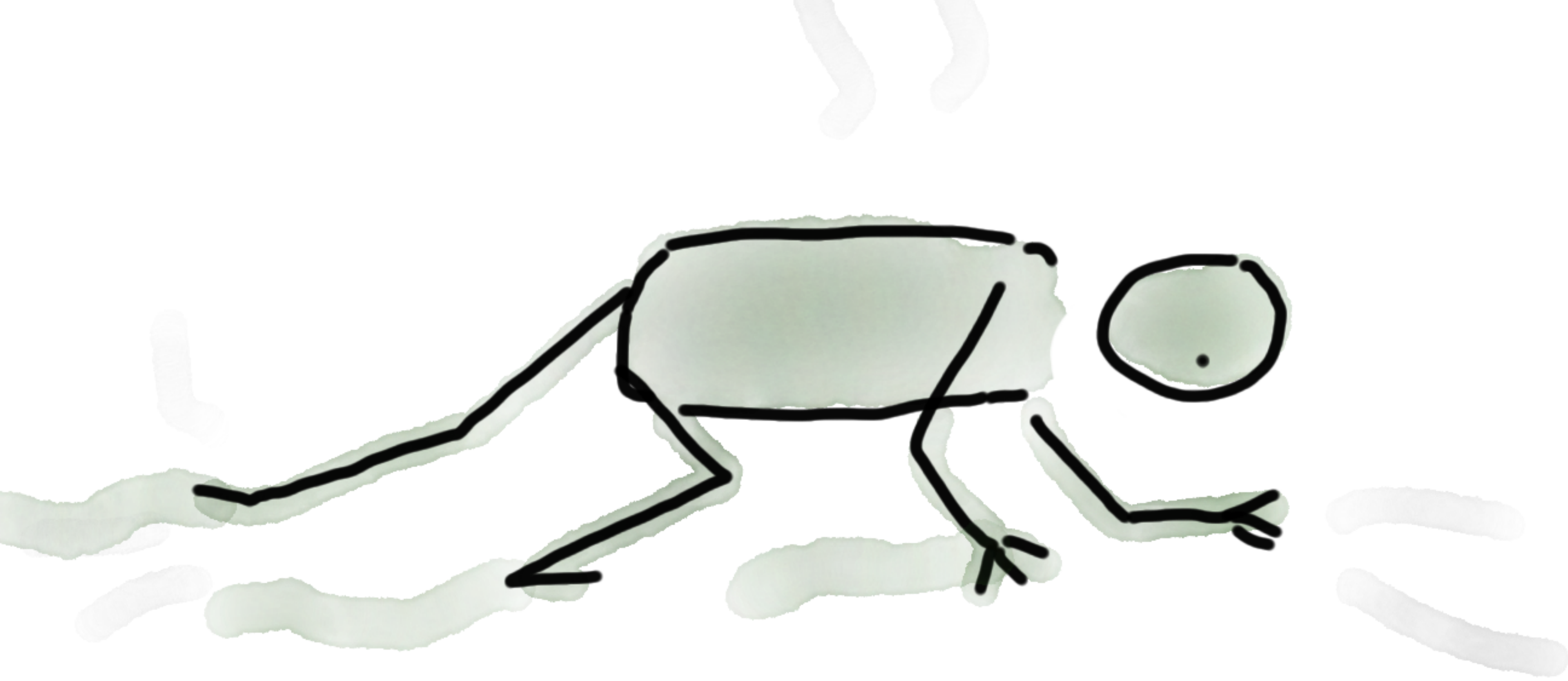


# zombie

“they haven't delivered any information or computing services for six months or more”

"comatose servers"





under-utilised servers

“much of the energy consumed by U.S. data centers is used to power more than 12 million servers that do little or no work most of the time”

NRDC



the average server:

12 - 18% of capacity

30 - 60 % of maximum power

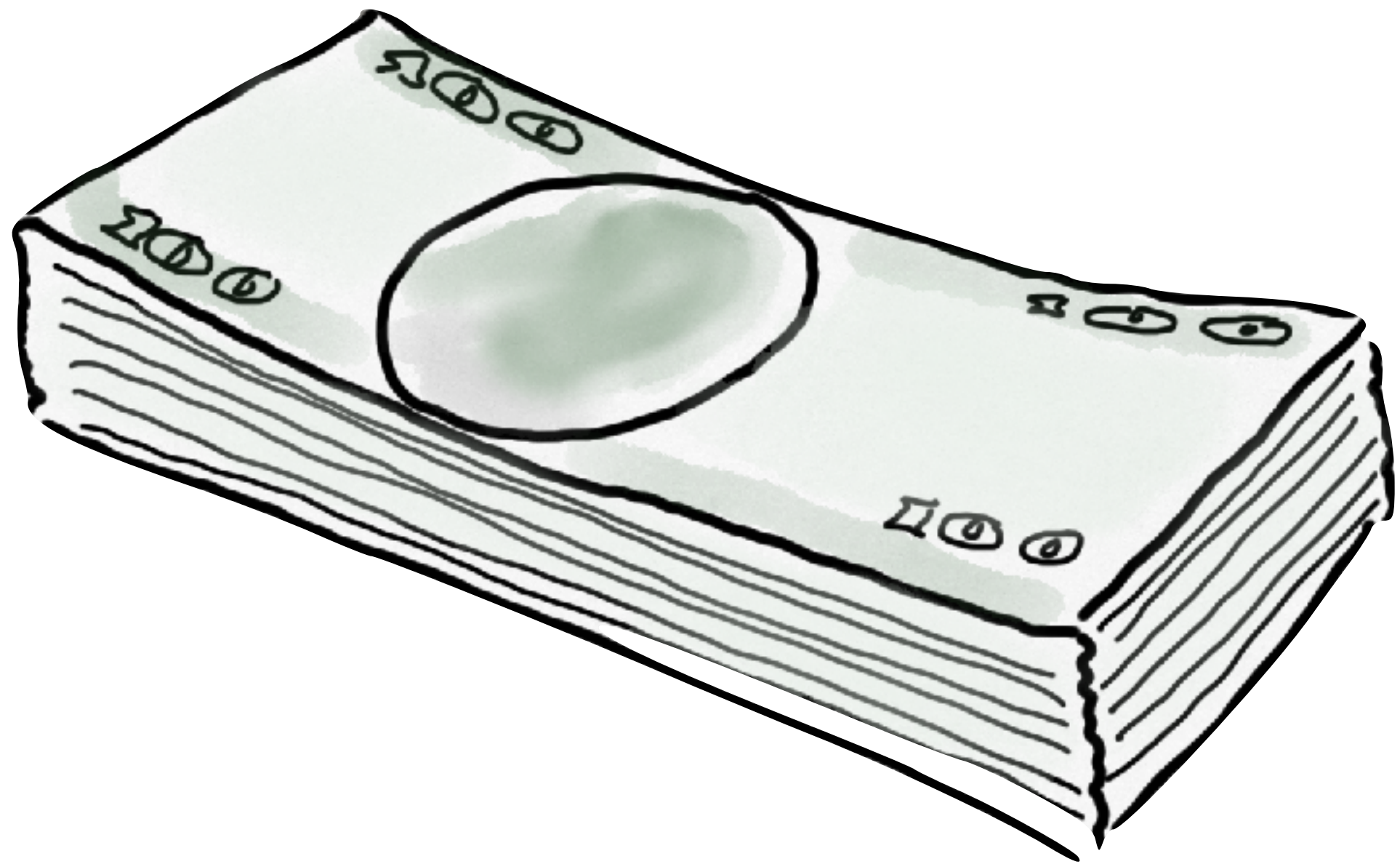
2014 survey

29%

of 4,000 active less than  
5% of the time



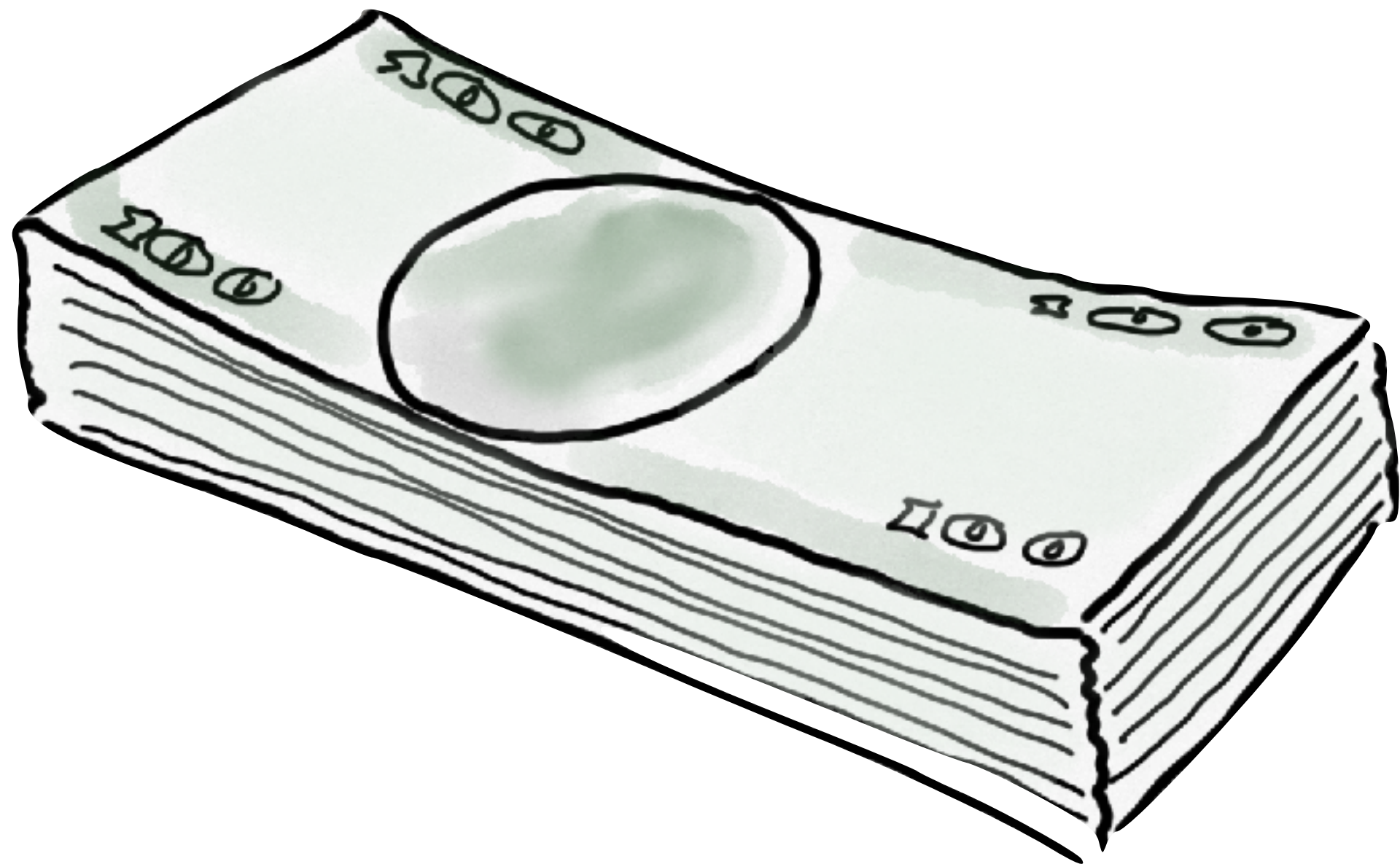
# 2021 study



<https://www.business2community.com/cloud-computing/overprovisioning-always-on-resources-lead-to-26-6-billion-in-public-cloud-waste-expected-in-2021-02381033>

2021 study

\$26.6 billion

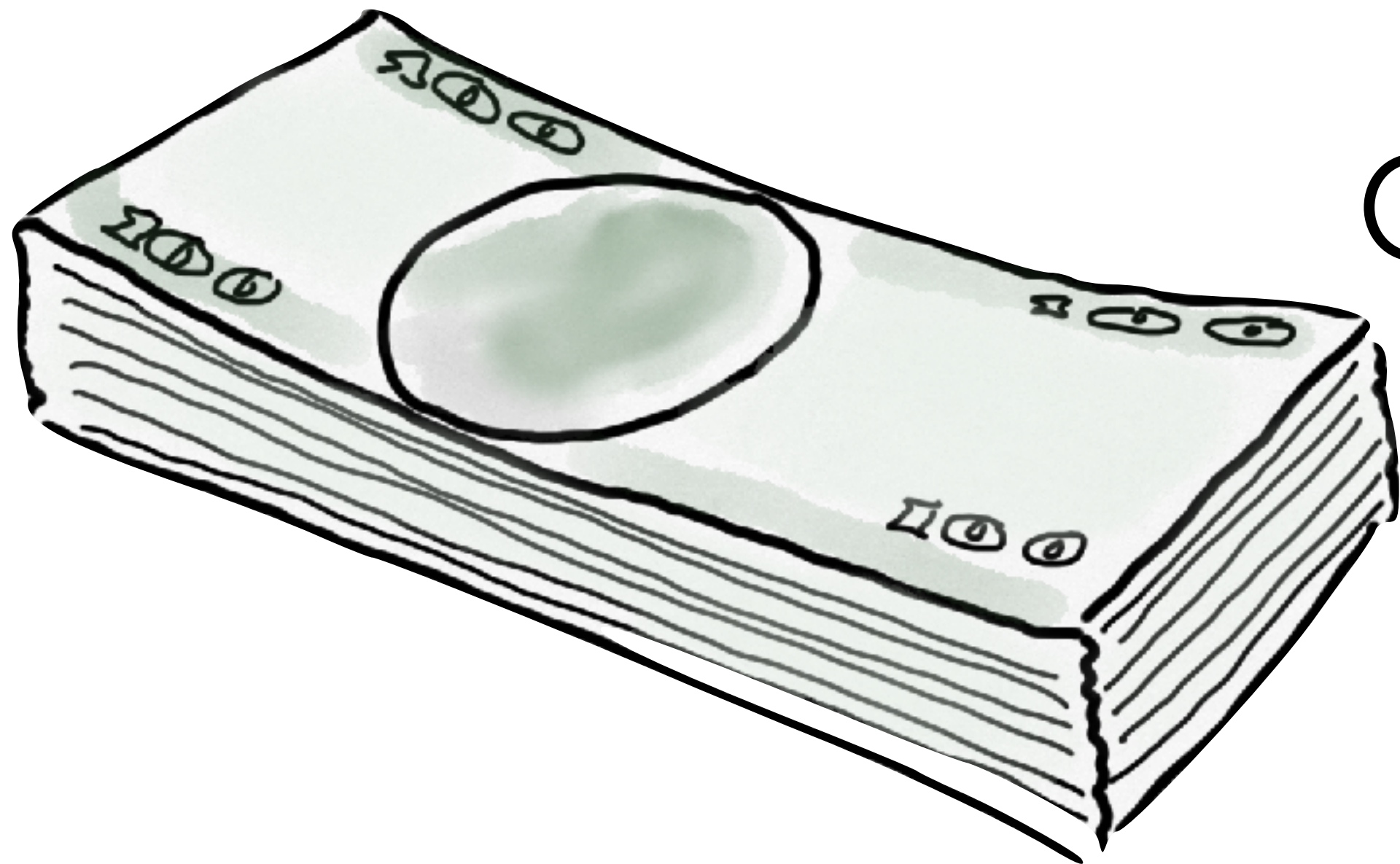


<https://www.business2community.com/cloud-computing/overprovisioning-always-on-resources-lead-to-26-6-billion-in-public-cloud-waste-expected-in-2021-02381033>

2021 study

\$26.6 billion

wasted by always-on  
cloud instances



<https://www.business2community.com/cloud-computing/overprovisioning-always-on-resources-lead-to-26-6-billion-in-public-cloud-waste-expected-in-2021-02381033>

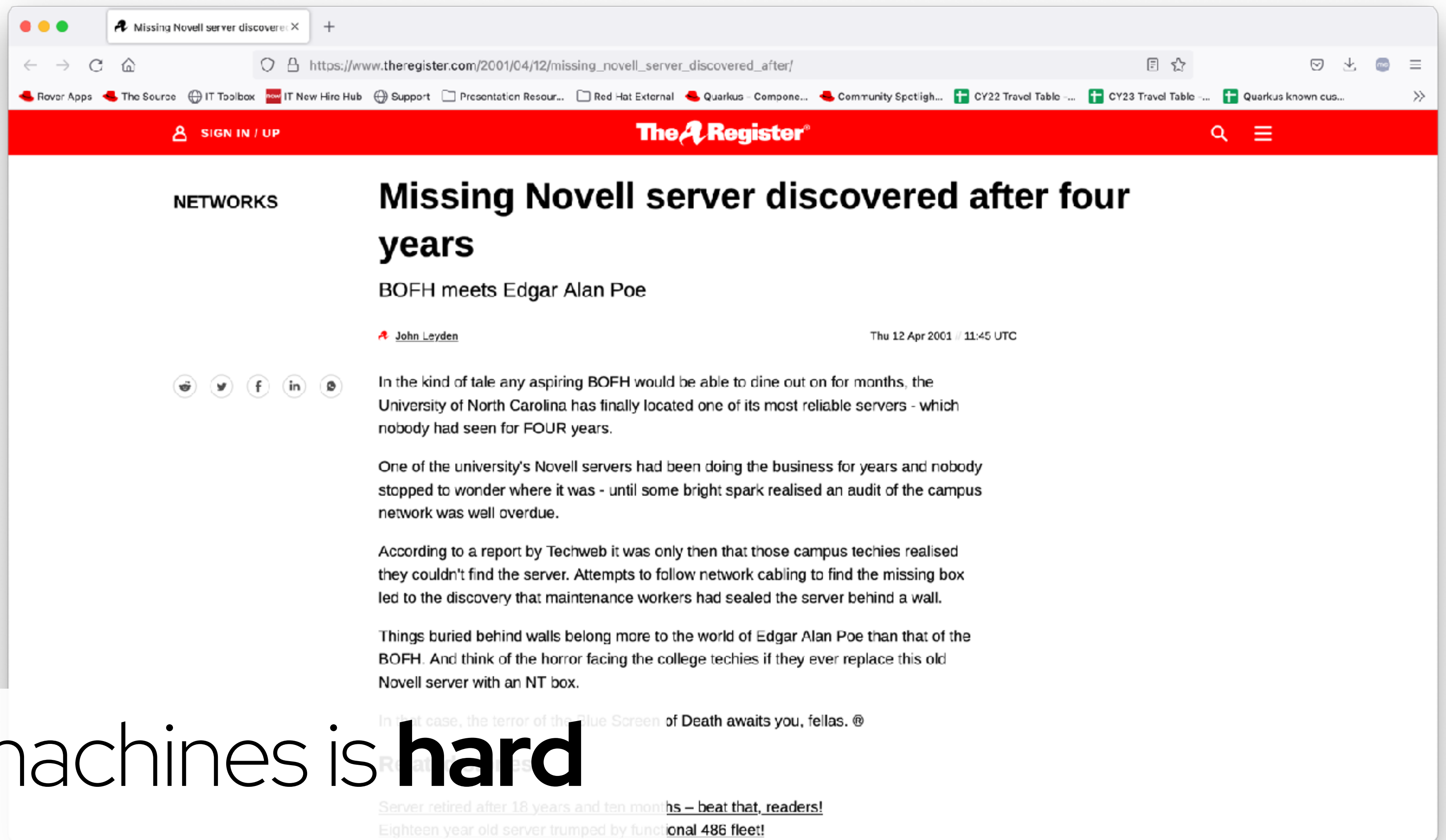
it's not just runtime costs

it's not just runtime costs

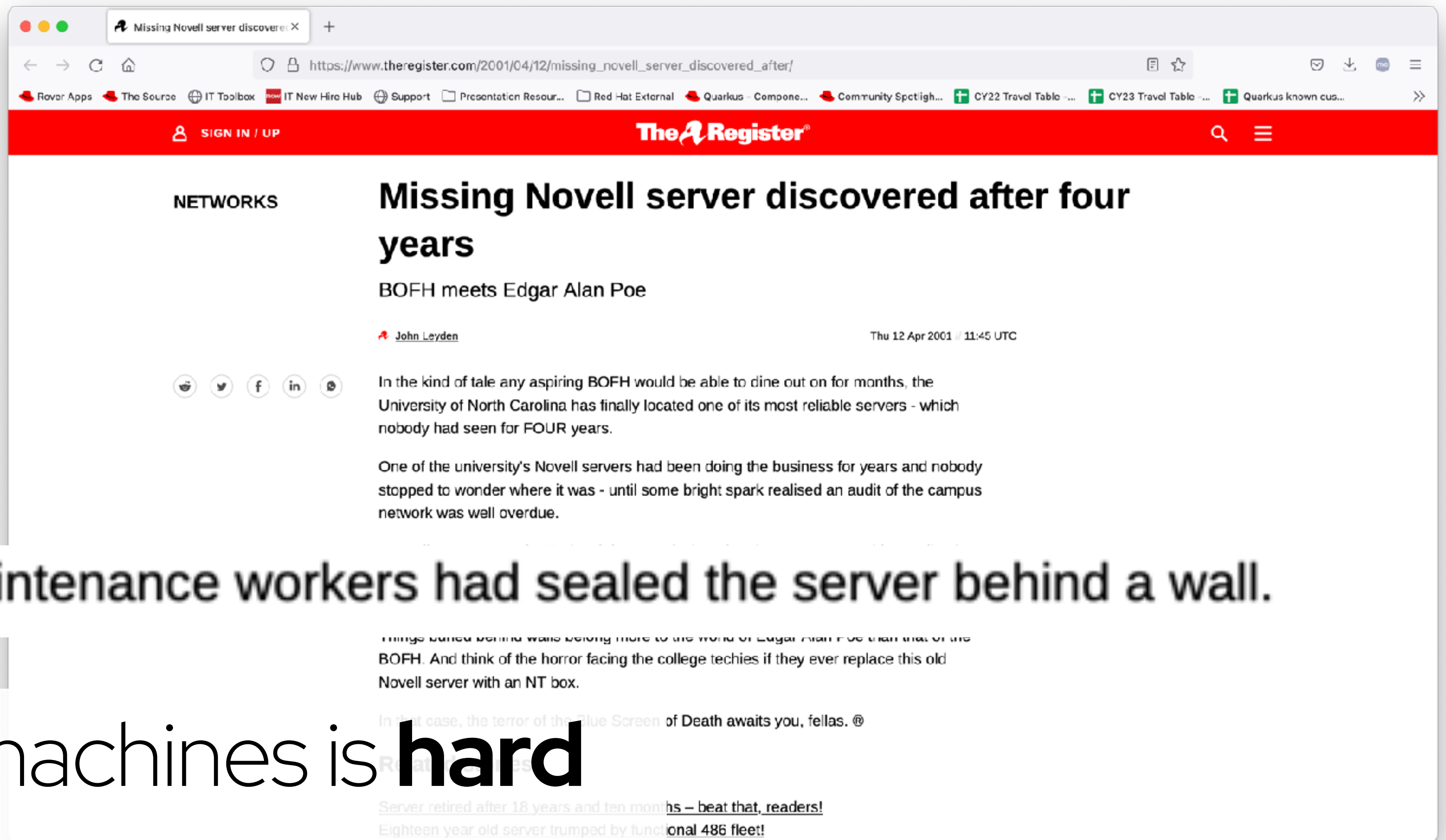
embodied carbon

why does this happen?





managing machines is **hard**



NETWORKS

# Missing Novell server discovered after four years

BOFH meets Edgar Allan Poe

John Leyden

Thu 12 Apr 2001 // 11:45 UTC



In the kind of tale any aspiring BOFH would be able to dine out on for months, the University of North Carolina has finally located one of its most reliable servers - which nobody had seen for FOUR years.

One of the university's Novell servers had been doing the business for years and nobody stopped to wonder where it was - until some bright spark realised an audit of the campus network was well overdue.

... maintenance workers had sealed the server behind a wall.

Things behind walls belong more to the world of Edgar Allan Poe than that of the BOFH. And think of the horror facing the college techies if they ever replace this old Novell server with an NT box.

In that case, the terror of the Blue Screen of Death awaits you, fellas. @

Server retired after 18 years and ten months – beat that, readers!

Eighteen year old server trumped by functional 486 fleet!

managing machines is **hard**





“perhaps someone  
forgot to turn them off”

Antithesis Institute

@holly\_cummins@hachyderm.io

#RedHat

projects ended

projects ended

business processes changed

projects ended

business processes changed

over-provisioning



projects ended

business processes changed

over-provisioning

isolation requirements

# risk averse processes



**Marcus Lyons** @marcuslyons\_ · Jan 12, 2022



Replying to @wesbos

We had at least 20 of those at my last job.

No one knew what they were for, took nearly 9 months of bureaucracy to finally shut them down



“we run this as a batch job on weekends”

“we run this as a batch job on weekends,  
but the servers stay up all week”

“we only use this system in UK working hours”

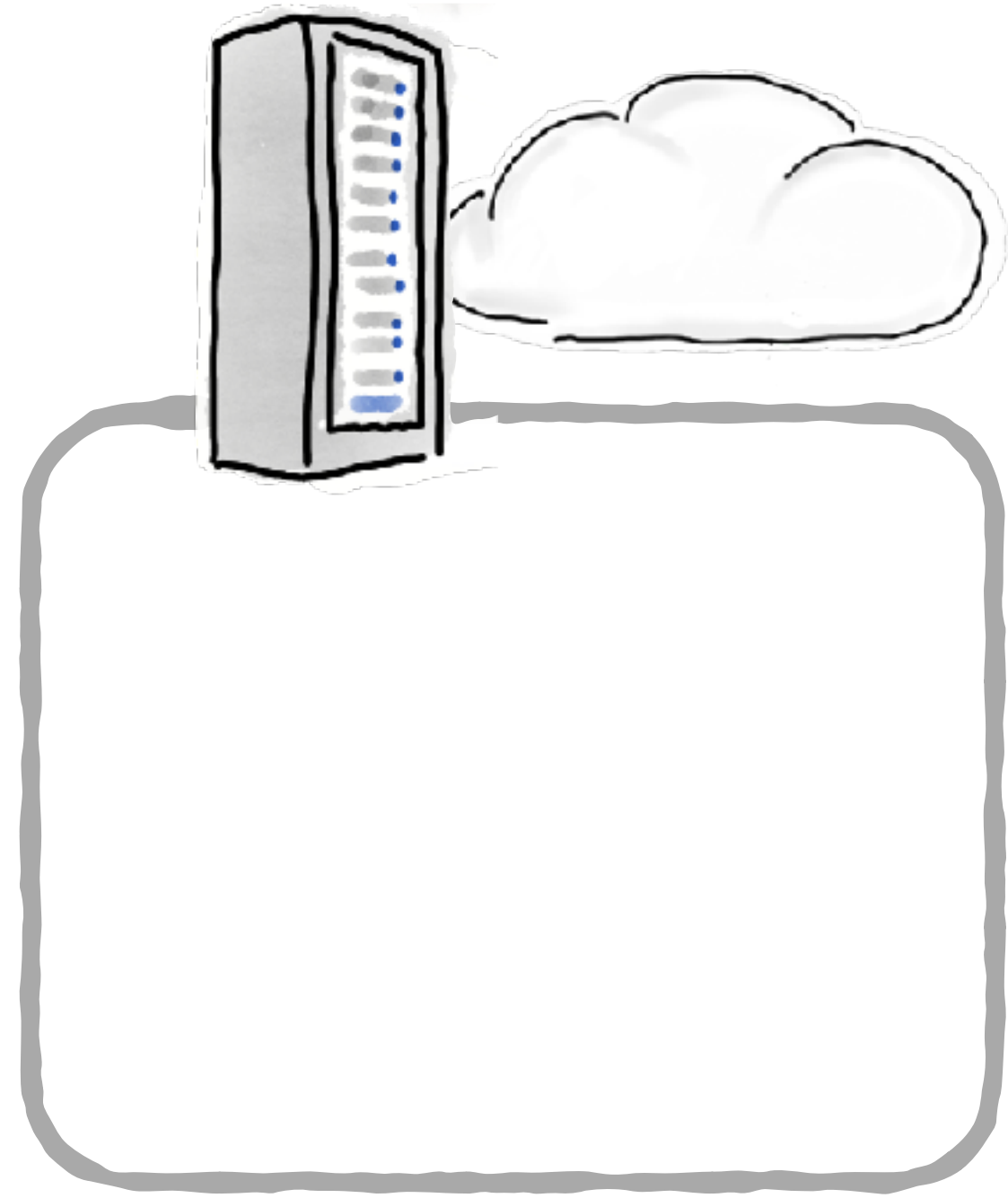
“we only use this system in UK working hours,  
but we leave it running 24/7 ”

auto-scaling algorithms are optimised for availability

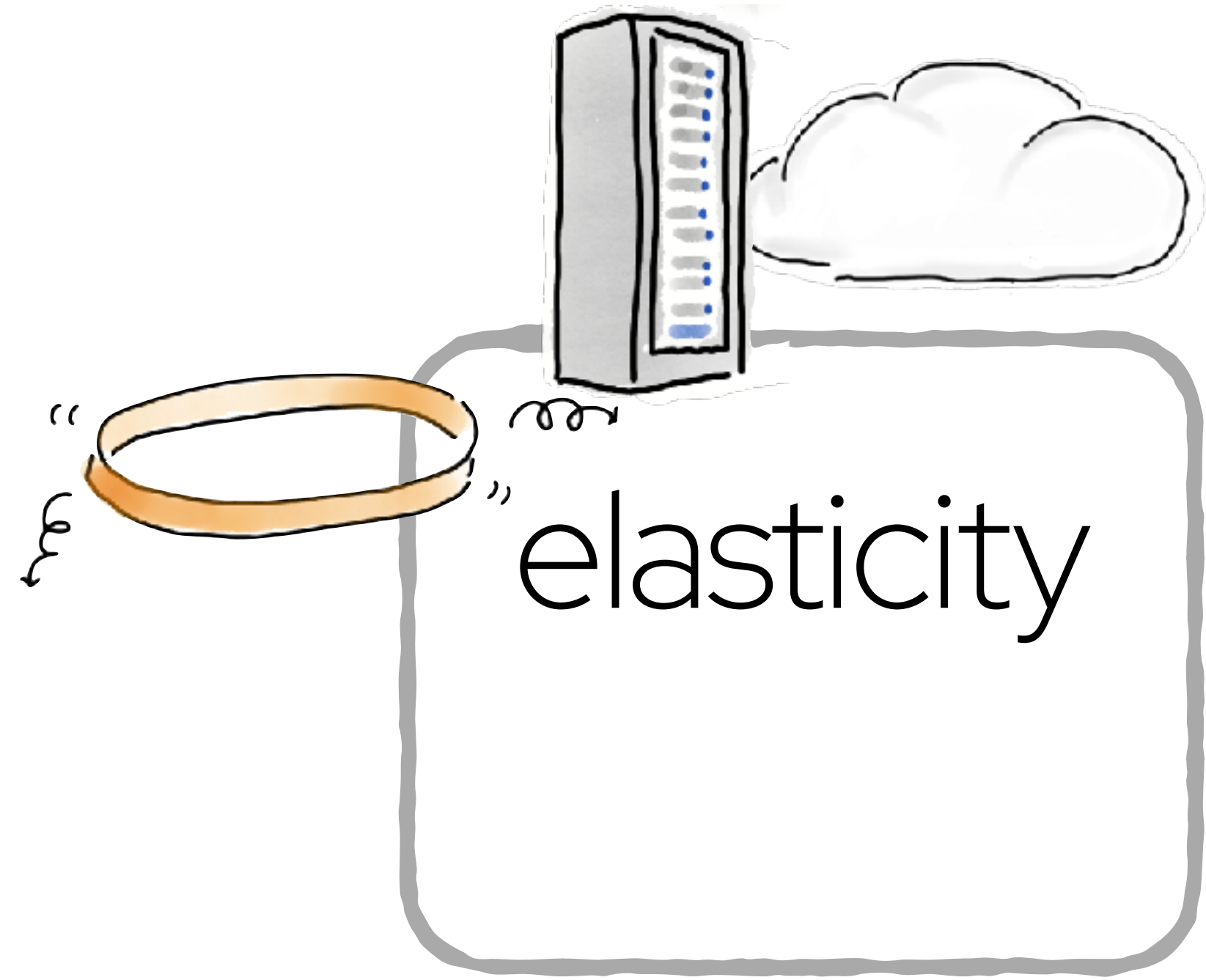
# green computing model: the four vowels



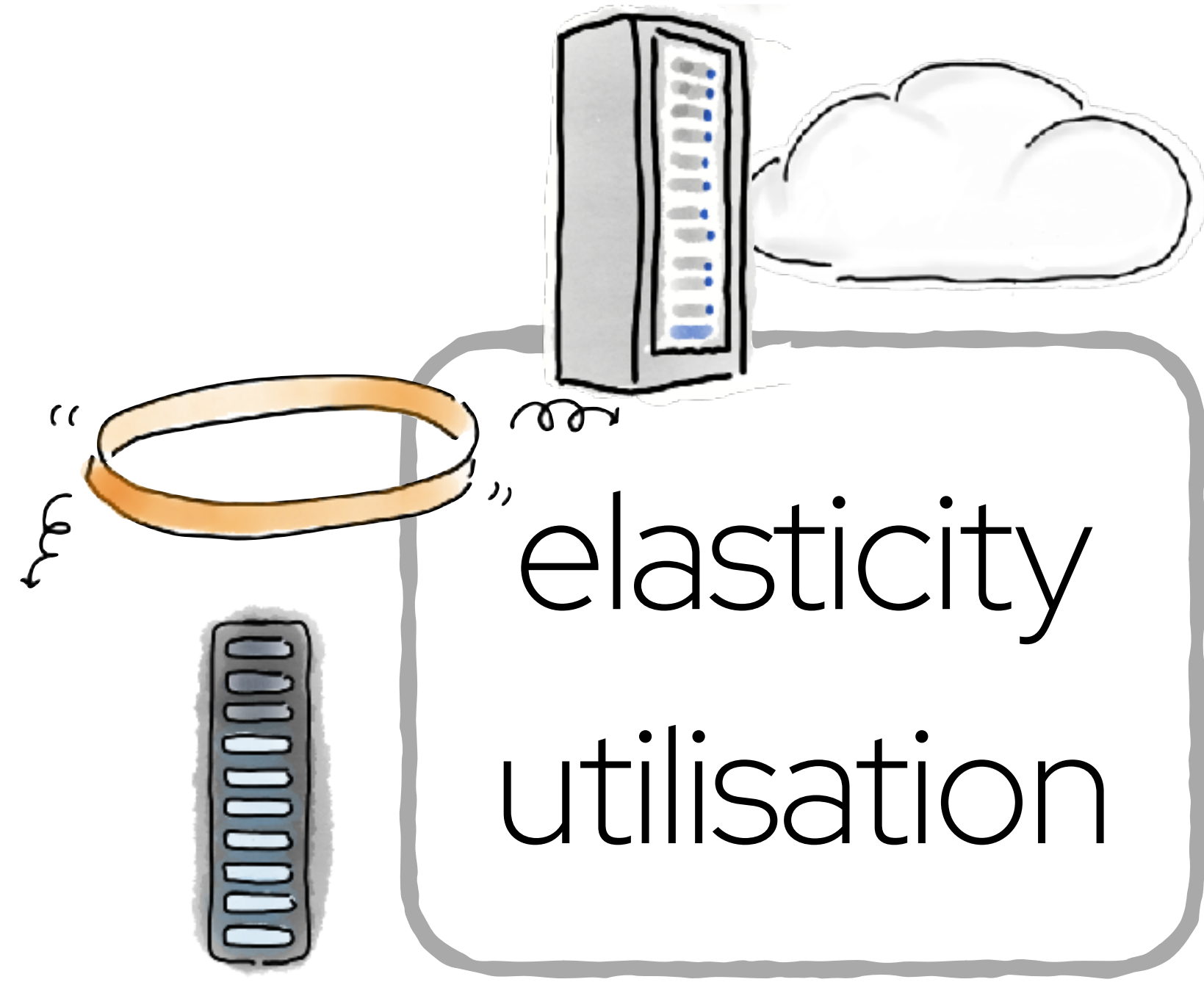
# green computing model: the four vowels



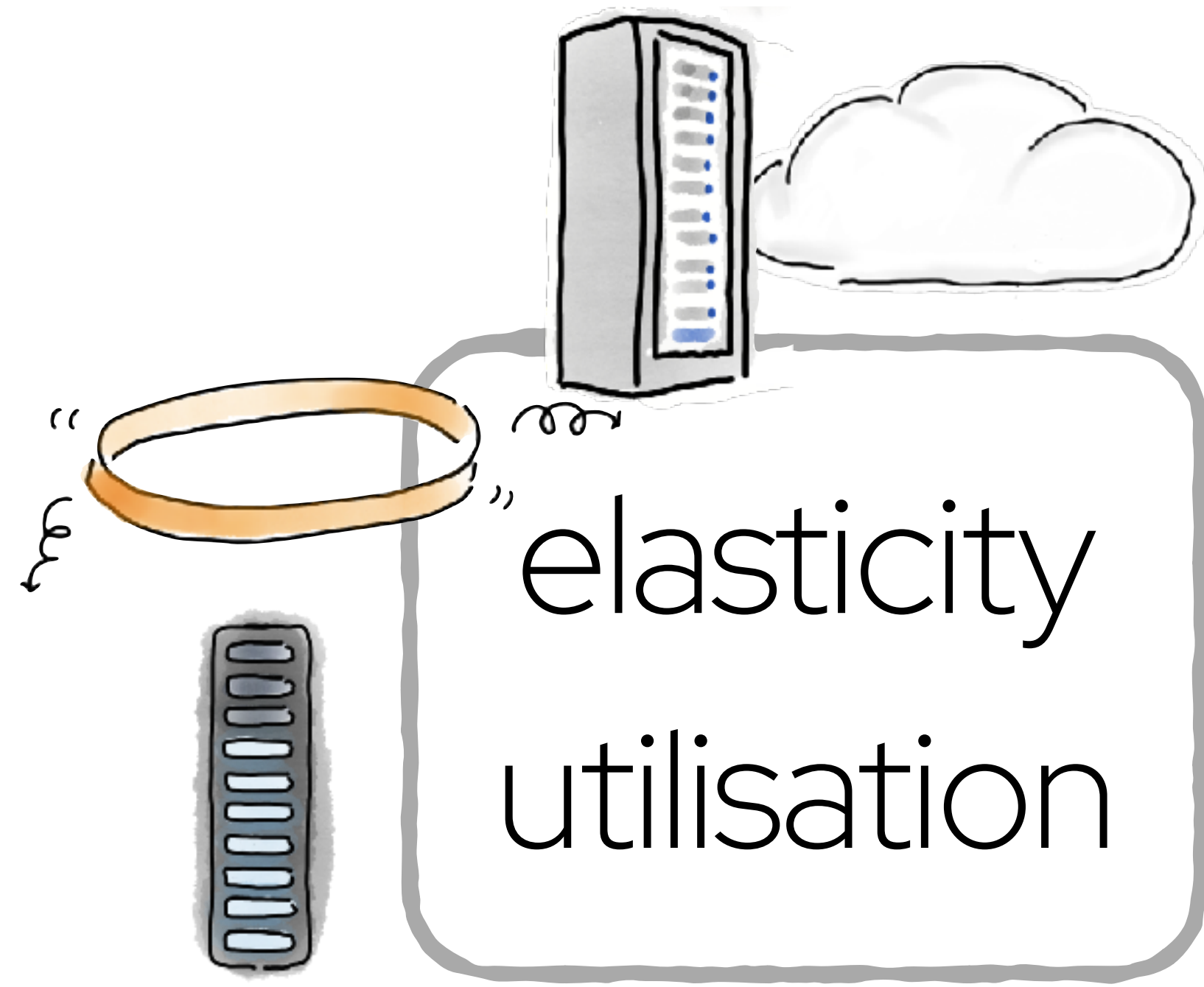
# green computing model: the four vowels



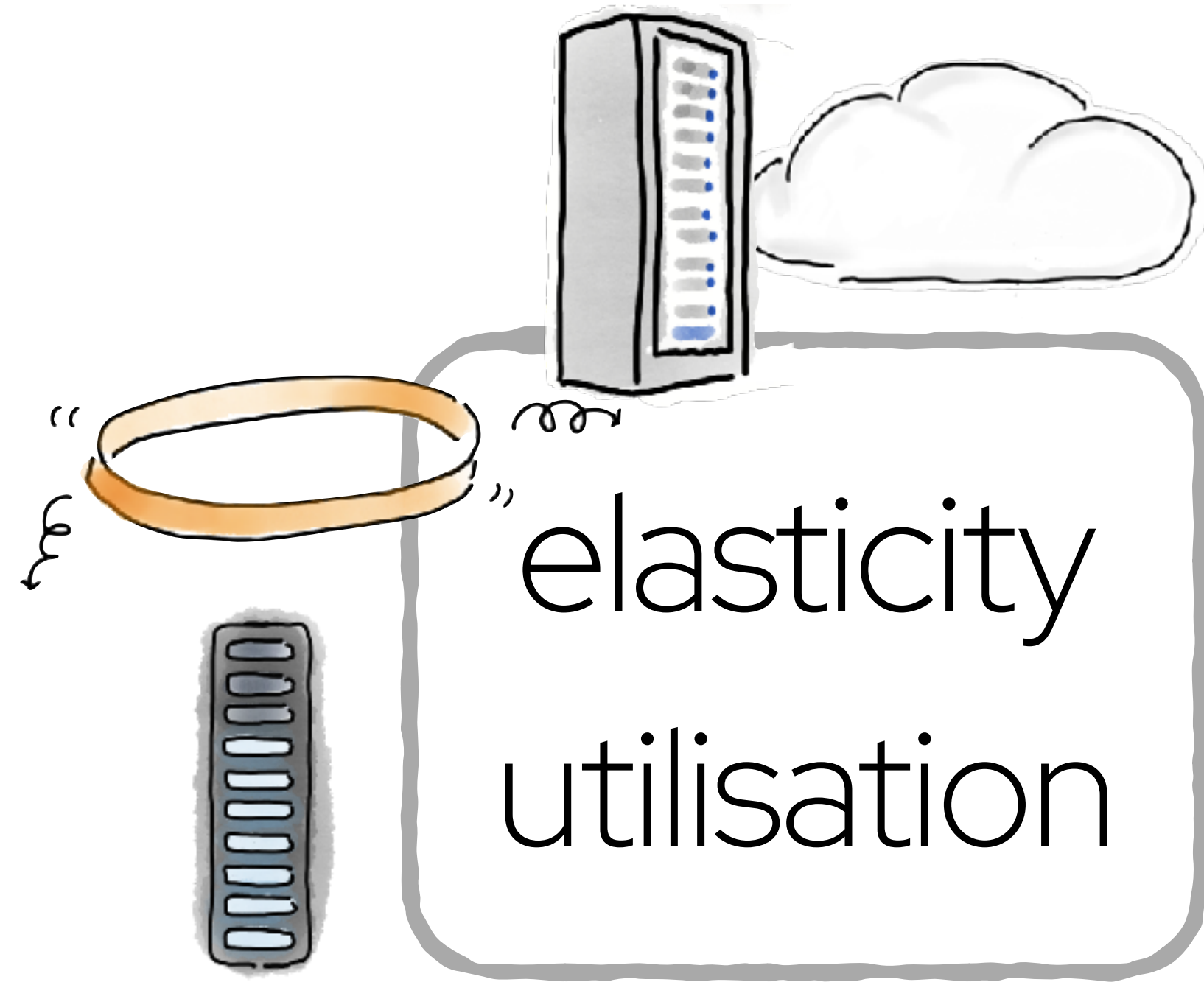
# green computing model: the four vowels



# green computing model: the four vowels

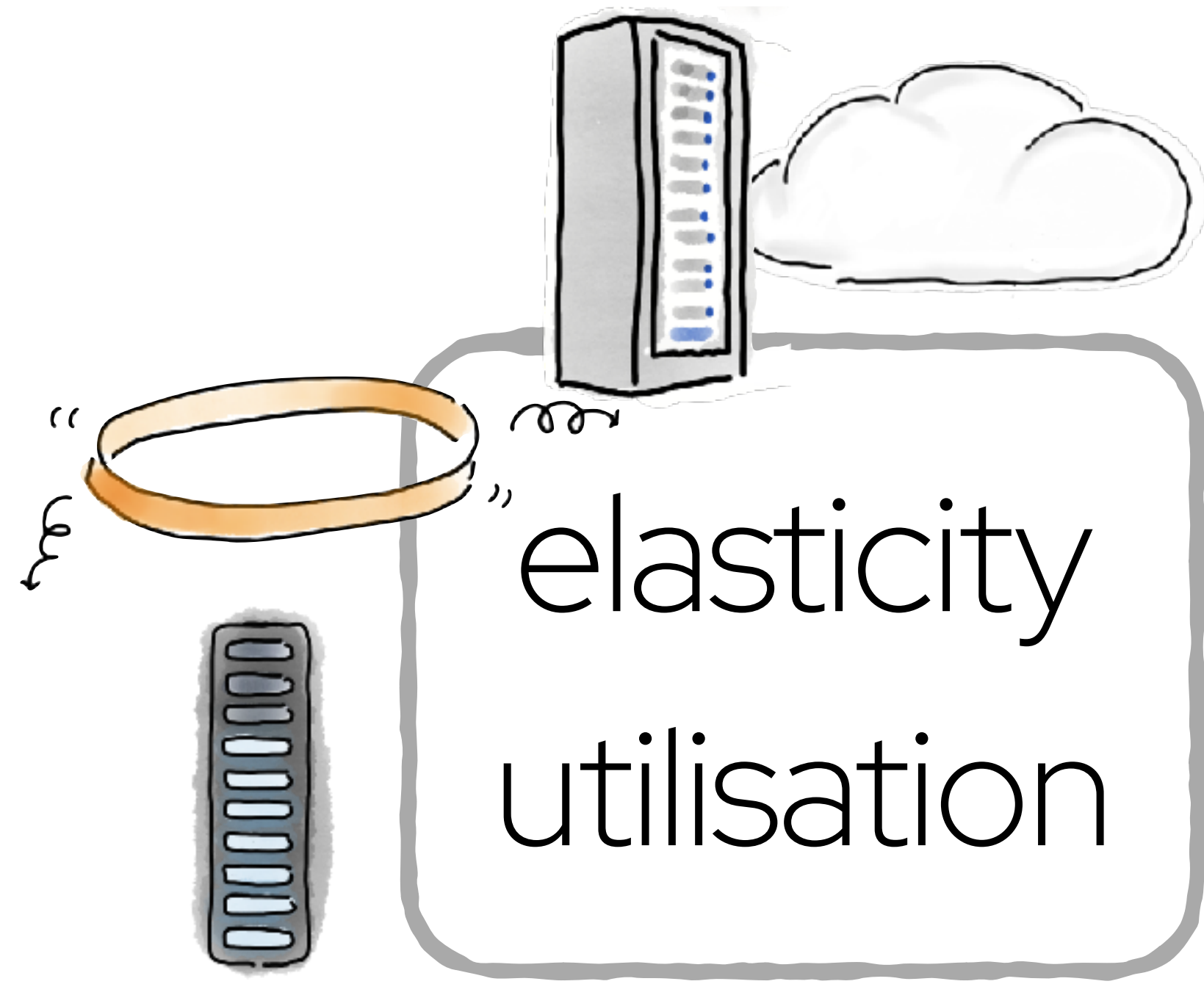


# green computing model: the four vowels

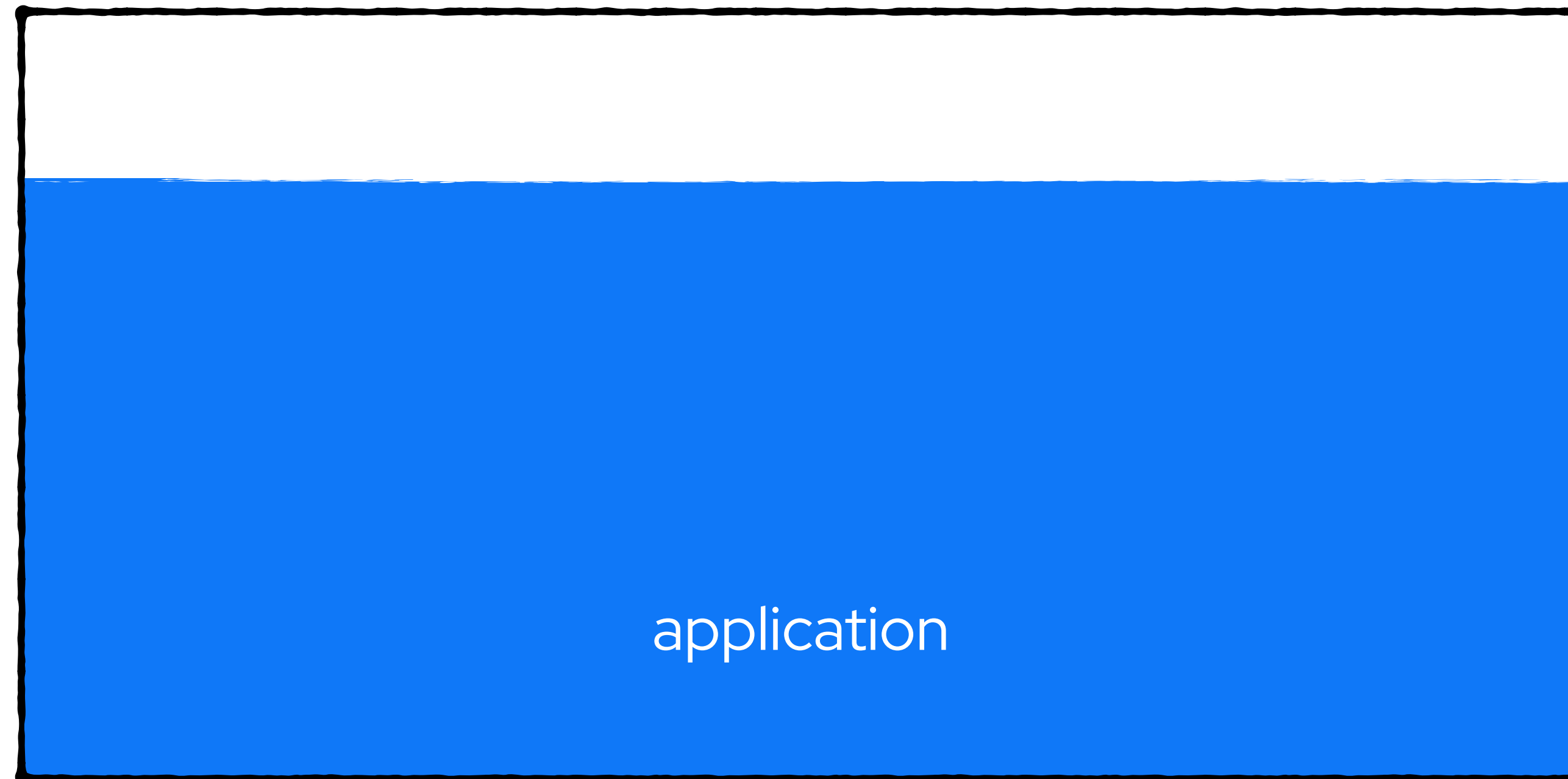


 utility

# green computing model: the four vowels

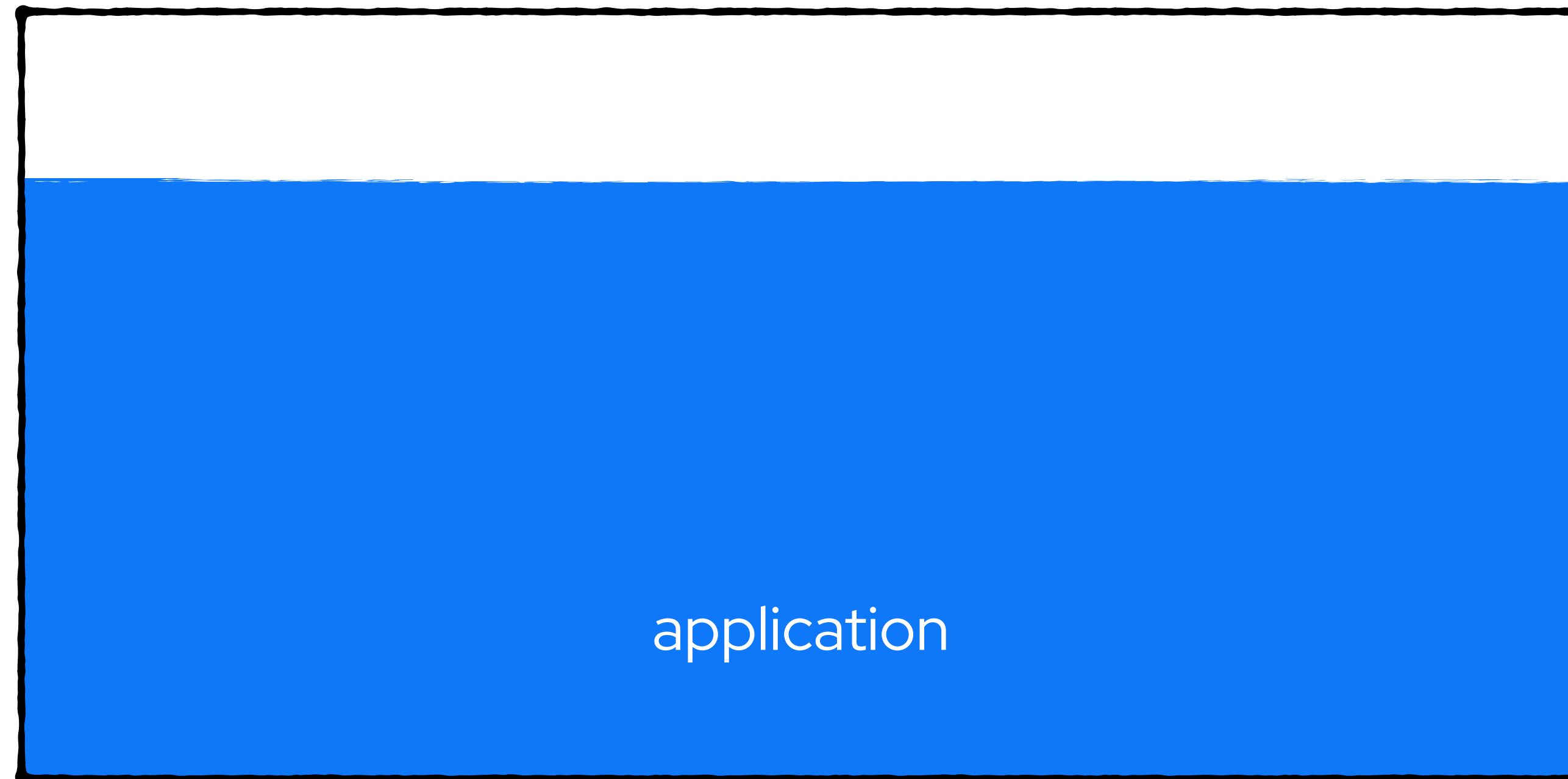


utilisation



# utilisation

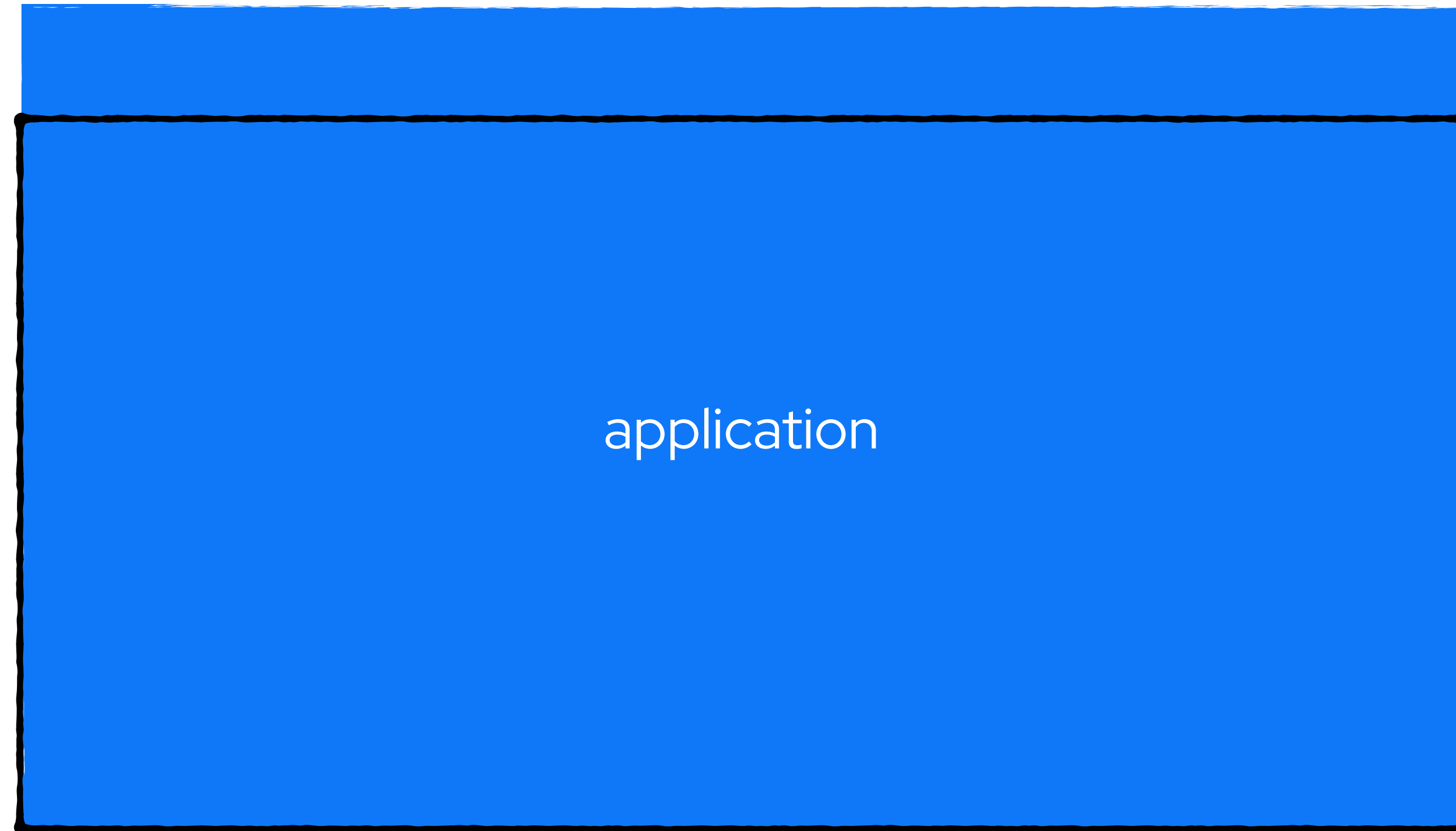
high utilisation  
good case





# utilisation

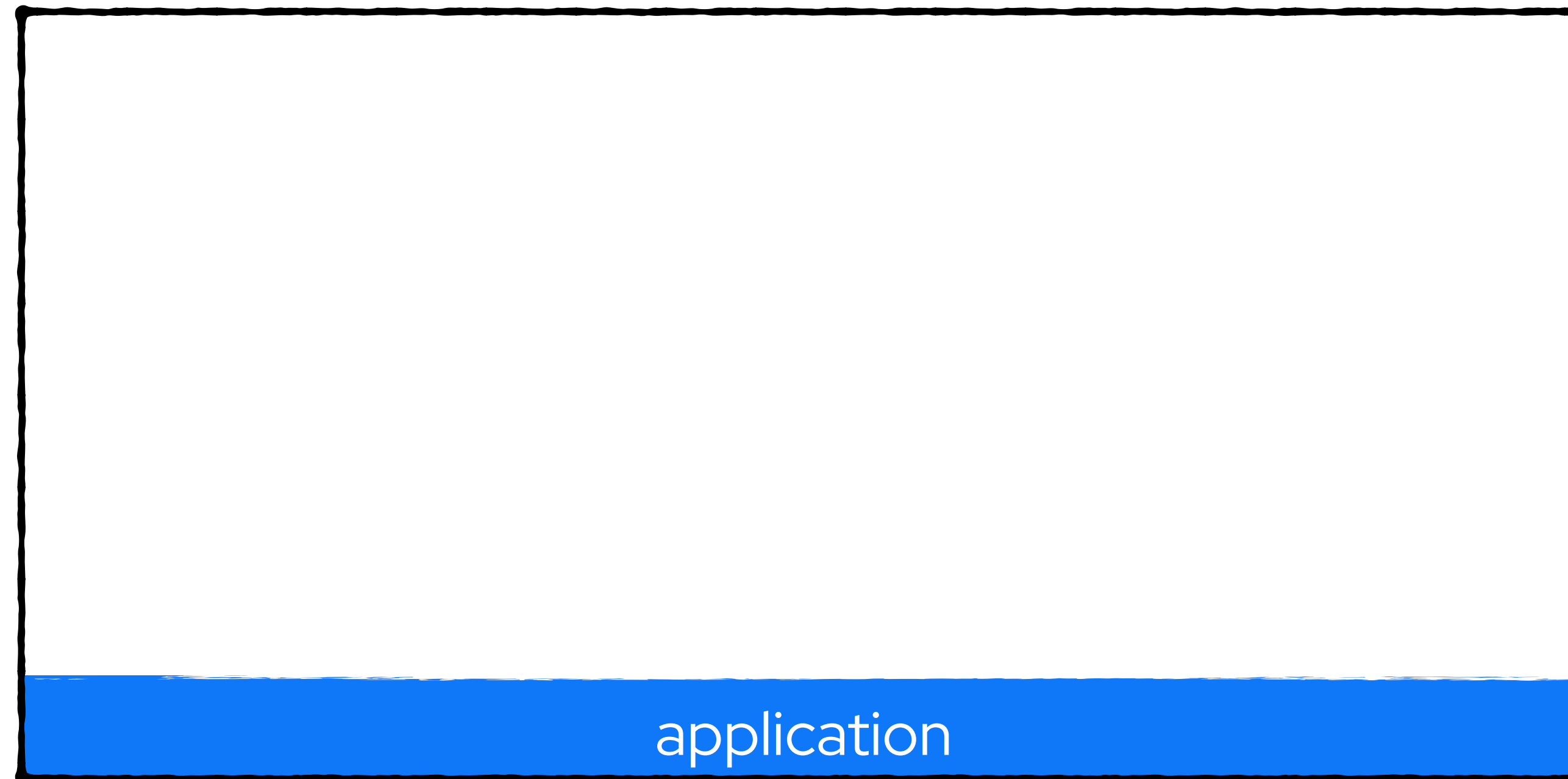
over-utilisation  
very bad case



# utilisation

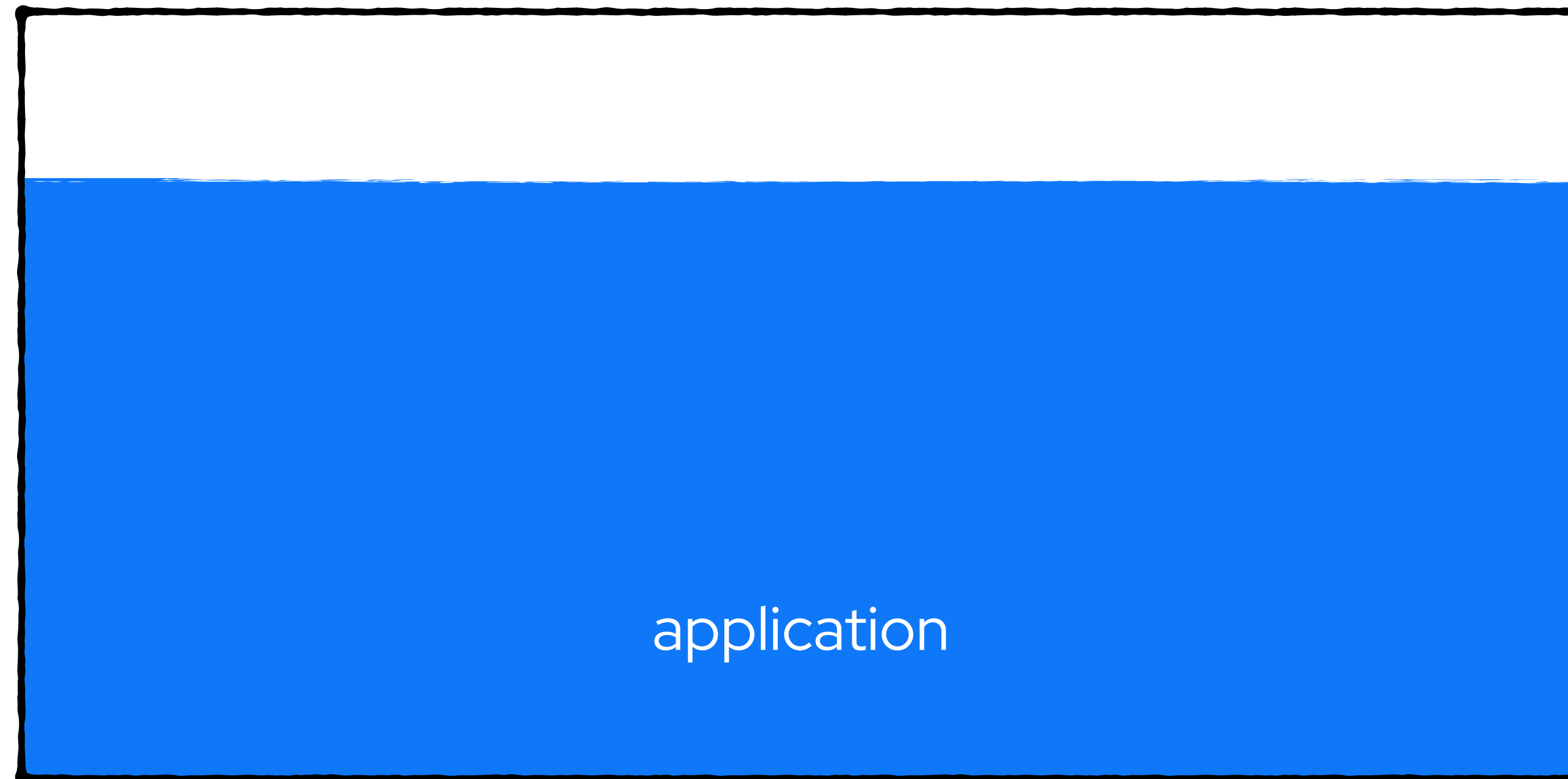
over-utilisation  
very bad case

under-utilisation  
wasteful case



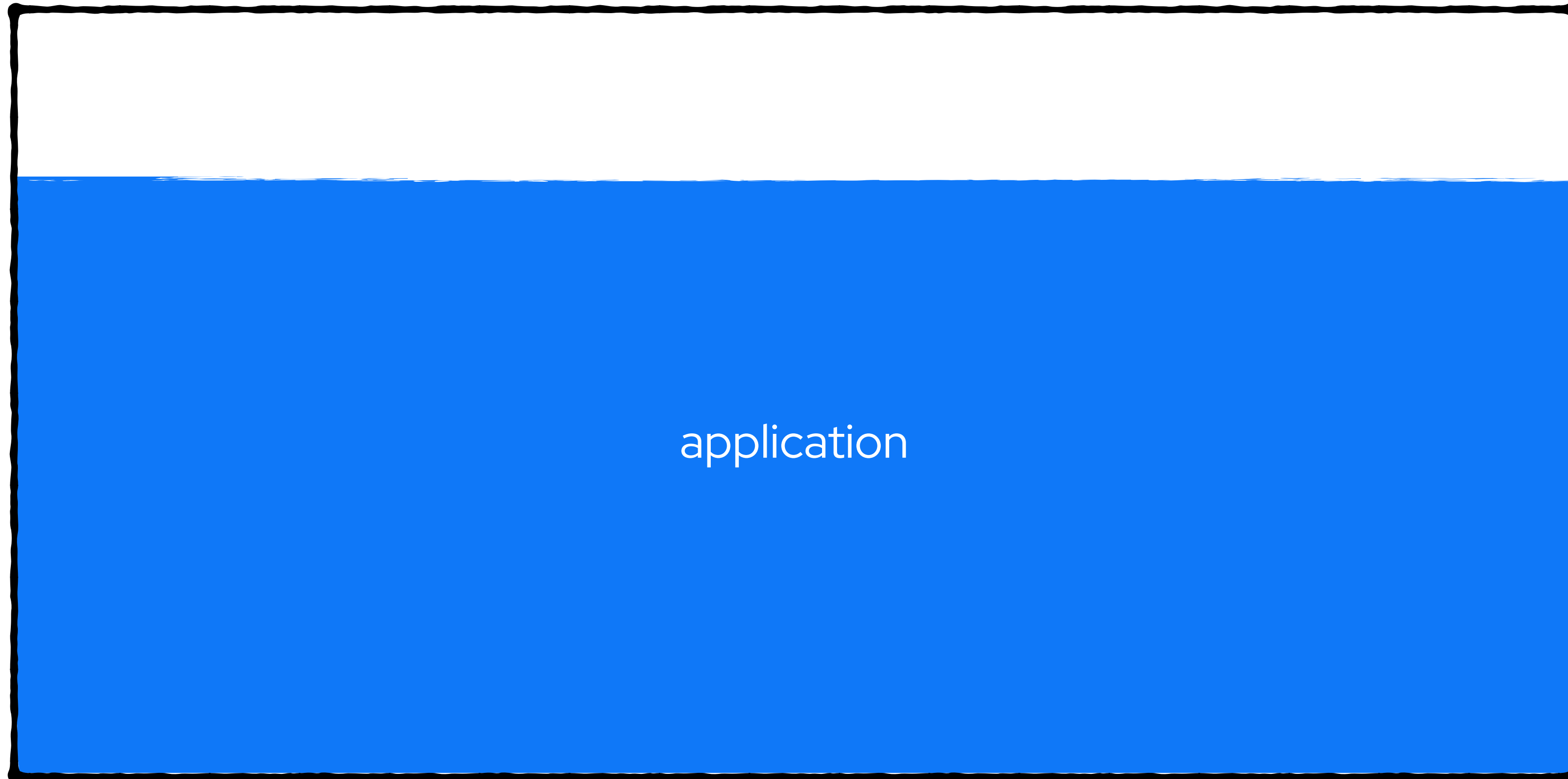
# elasticity

high utilisation  
good case



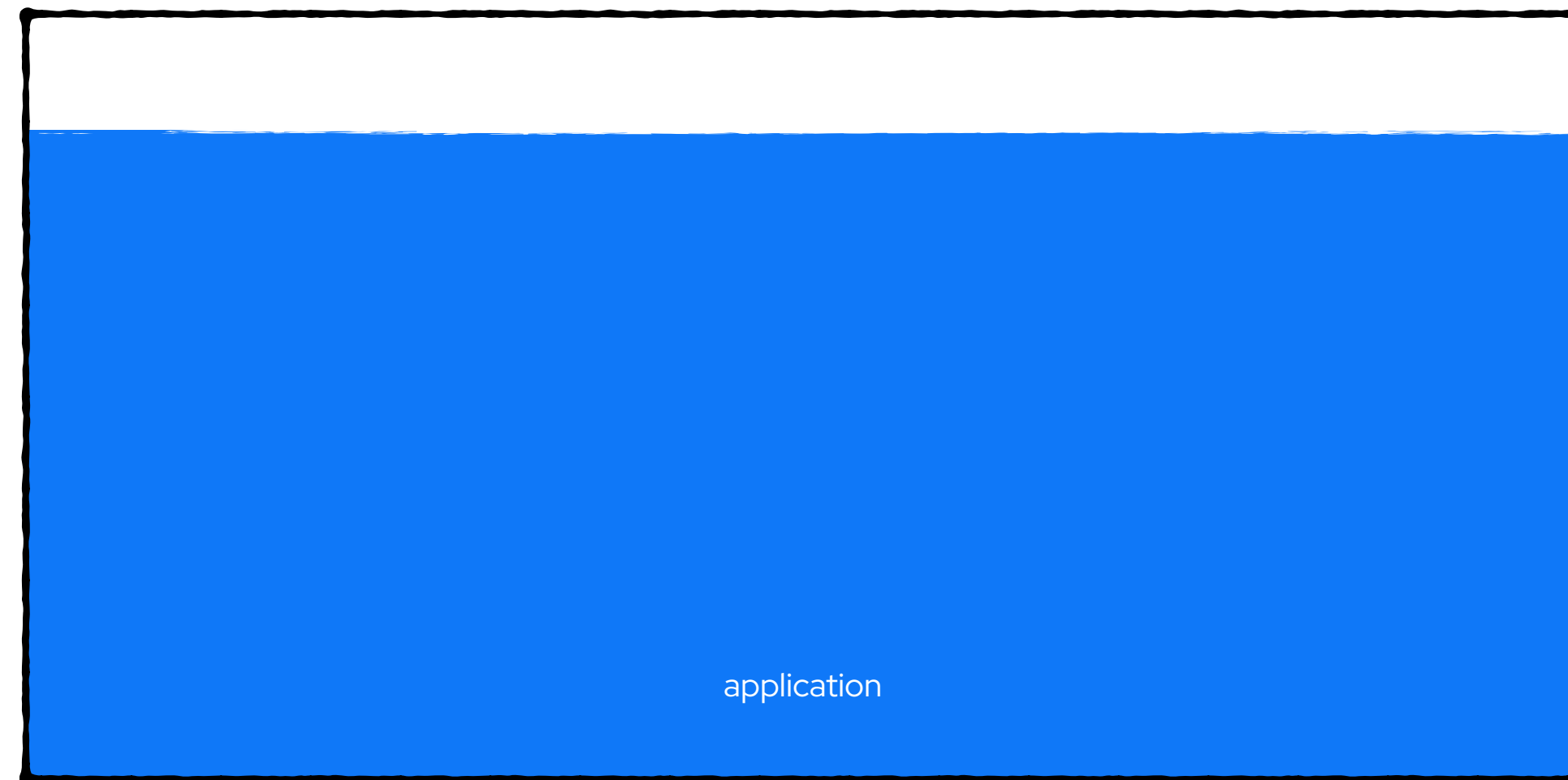
# elasticity

scale-up  
good utilisation



# elasticity

scale-down  
good utilisation



# green computing model: the four vowels



```
public class ... {  
    private thing ()  
    private another ... ()  
    ...  
}
```

efficiency



# green computing model: the four vowels



 utility

There is nothing so useless as  
doing efficiently that which  
should not be done at all.

Peter Drucker




"efficient zombies"

how do we solve the zombie problem?

how do we solve the zombie problem?

detection and destruction

[Subscribe to Digest](#)[Account](#) 

Arts & Life

## Top 5: Ways to kill a zombie

By **Intermission Staff**  
Oct. 21, 2011, 12:35 a.m.

To commemorate the second season premiere of “**The Walking Dead**” on AMC last Sunday, the cast selected their choice zombie-slaying tools at New York Comic-Con. We here at Intermission aren’t sure if we’re ready to live a life of secluded Twinkie-eating and cockroach-befriending quite yet, but just in case that darned zombie apocalypse pops up anytime soon, here’s how we’d deal with those undead suckers.

### **Eternal flamethrower**

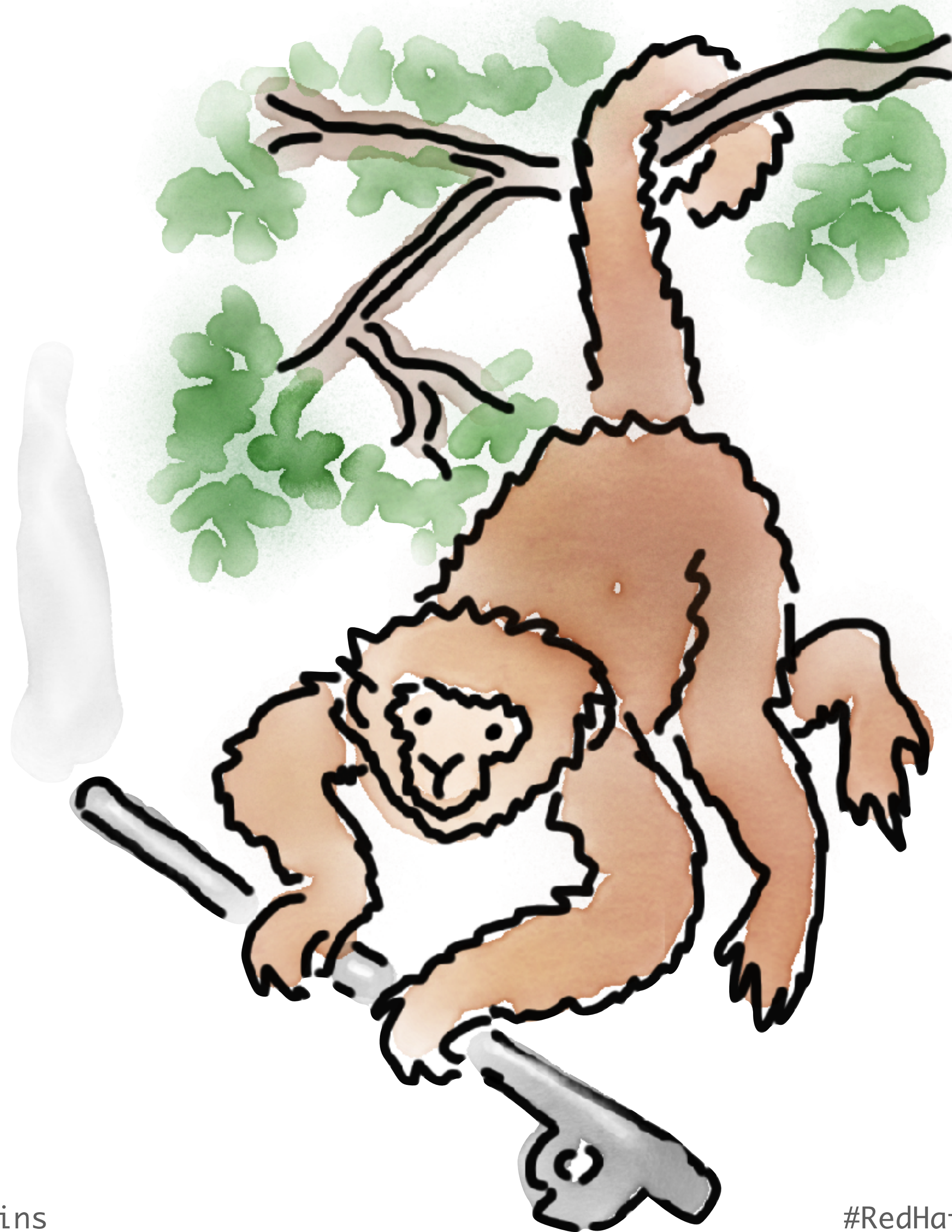
We’re fairly certain we’ve never seen a George A. Romero flick with fire-retardant zombies, so this is a pretty safe bet. Throw in some sort of technological innovation to keep the flame going and it’s the gift that keeps on giving.



system archaeology

... is not easy

scream test



"eco-monkey"

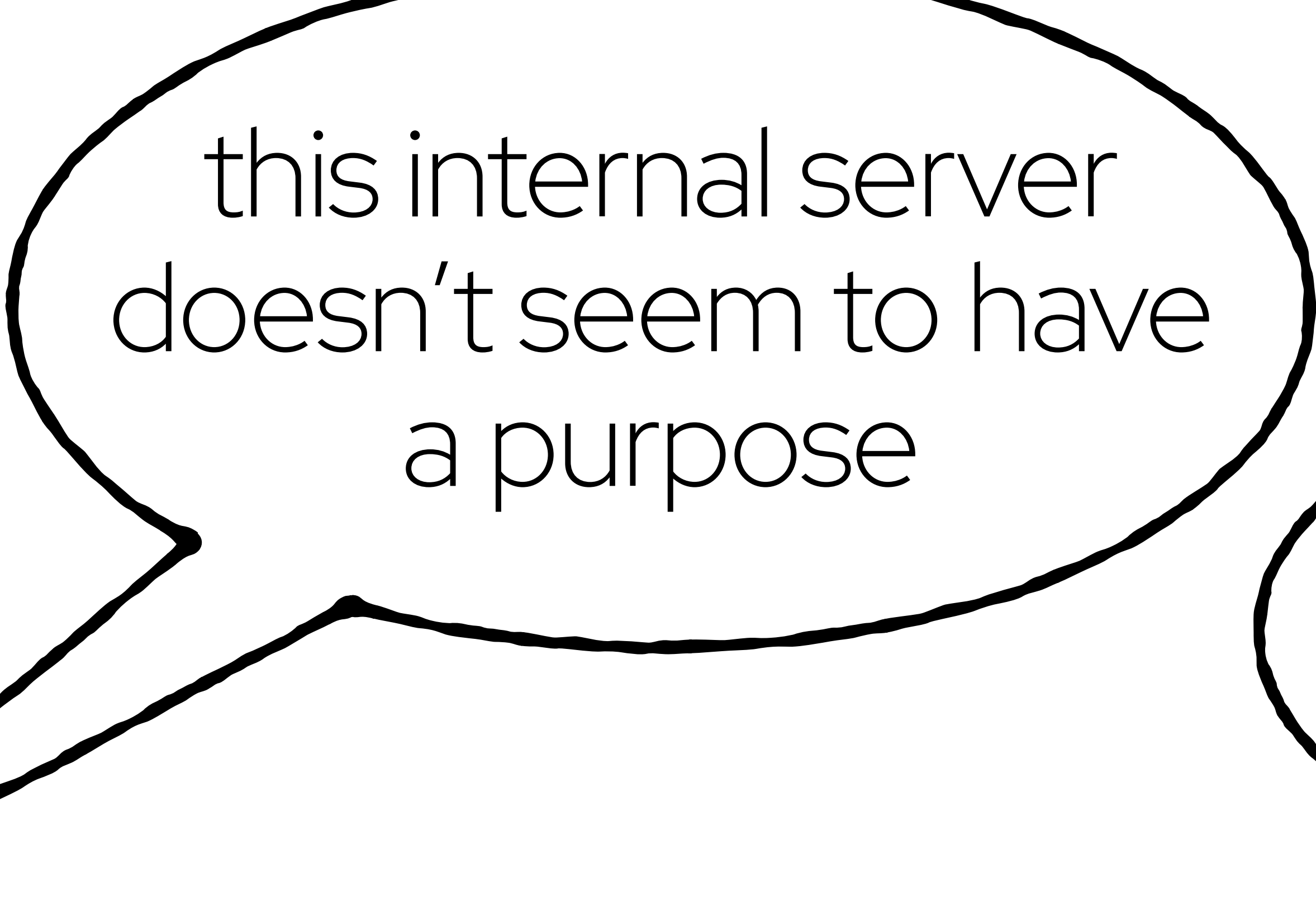
the scream is real





this internal server  
doesn't seem to have  
a purpose

the scream is real



this internal server  
doesn't seem to have  
a purpose

the scream is real



let's turn it off!

this internal server  
doesn't seem to have  
a purpose

the scream is real

let's turn it off!

uh ... why did the  
backbone of a  
client's network  
just vanish?

this internal server  
doesn't seem to have  
a purpose

the scream is real

let's turn it off!

uh ... why did the  
backbone of a  
client's network  
just vanish?

*oops.*

# long meetings

let's figure out what all these cloud workloads are, since I'm **paying** for them



IT Department, UK Bank

# long meetings

let's figure out what all these cloud workloads are, since I'm **paying** for them



IT Department, UK Bank

Zombies!

to: alice@acme.org,  
bob@acme.org

Zombies!

hi all,  
Zombies are on the  
rampage in our org.  
please check your  
servers to see if they  
are undead or maybe  
also look for vampires.

CFO Chuck

long emails

tags

holly x

delete-94 x

dev x

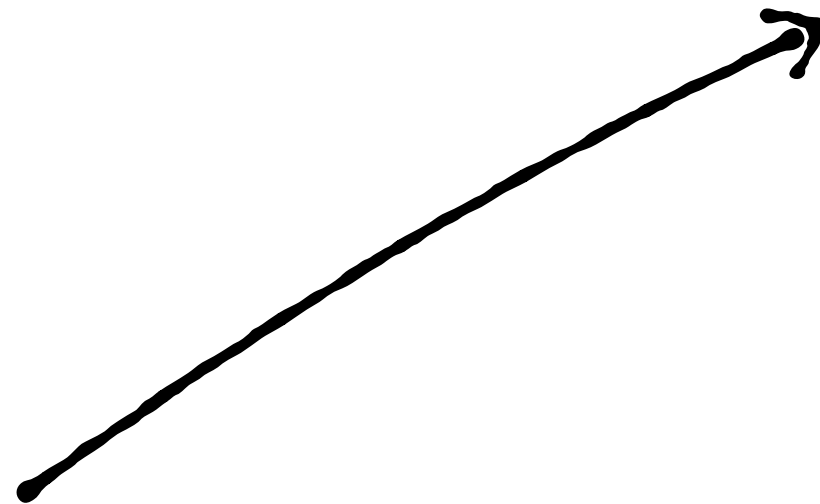


all the –opses

# GreenOps

# GreenOps

greenops is a mid-sized trilobite (really)



# FinOps

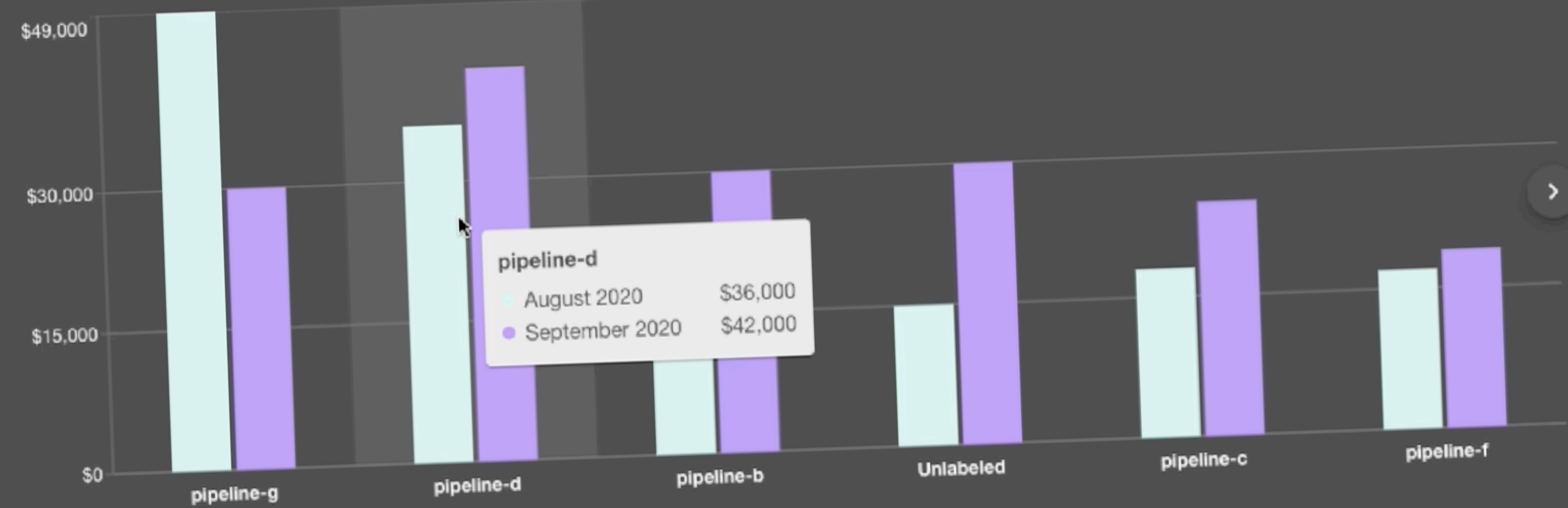
figuring out who in your company forgot to turn off their cloud

# Data Processing

9 entities, sorted by cost

August vs September ▾

● AUGUST 2020 ● SEPTEMBER 2020 COST GROWTH  
\$200,000 \$250,000 20% or ~3 engineers



**pipeline-d**

August 2020	\$36,000
September 2020	\$42,000

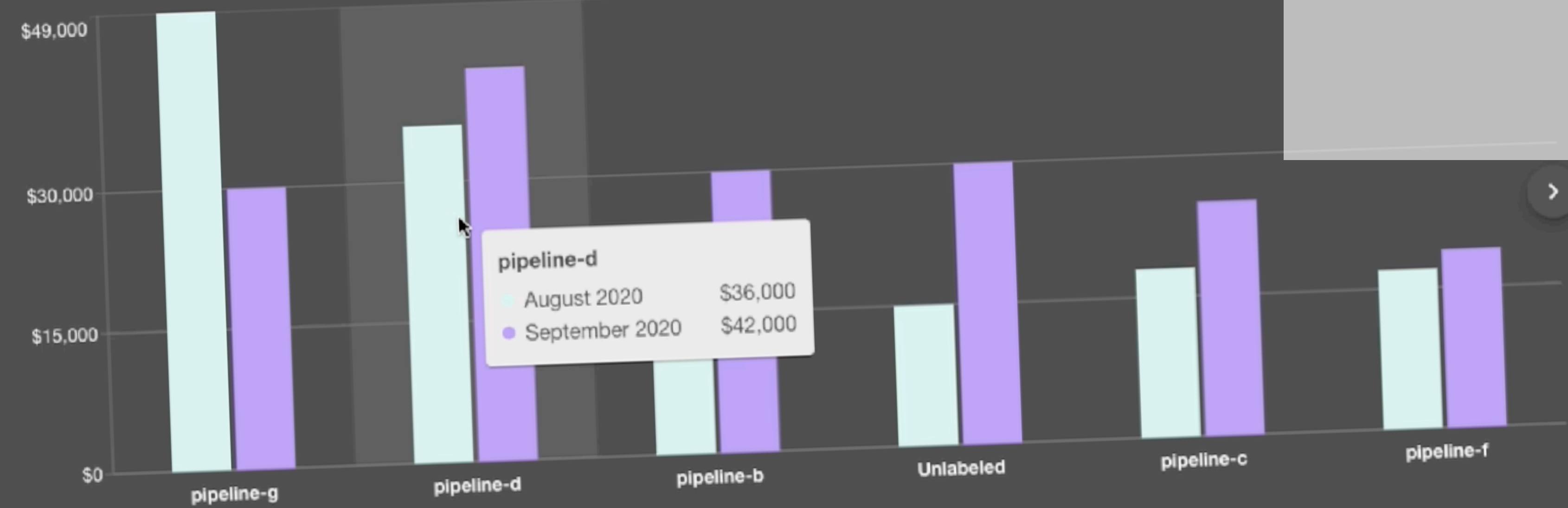
# Data Processing

9 entities, sorted by cost

August vs September ▾

● AUGUST 2020 ● SEPTEMBER 2020 COST GROWTH  
\$200,000 \$250,000 20% or ~3 engineers

backstage.io



**pipeline-d**

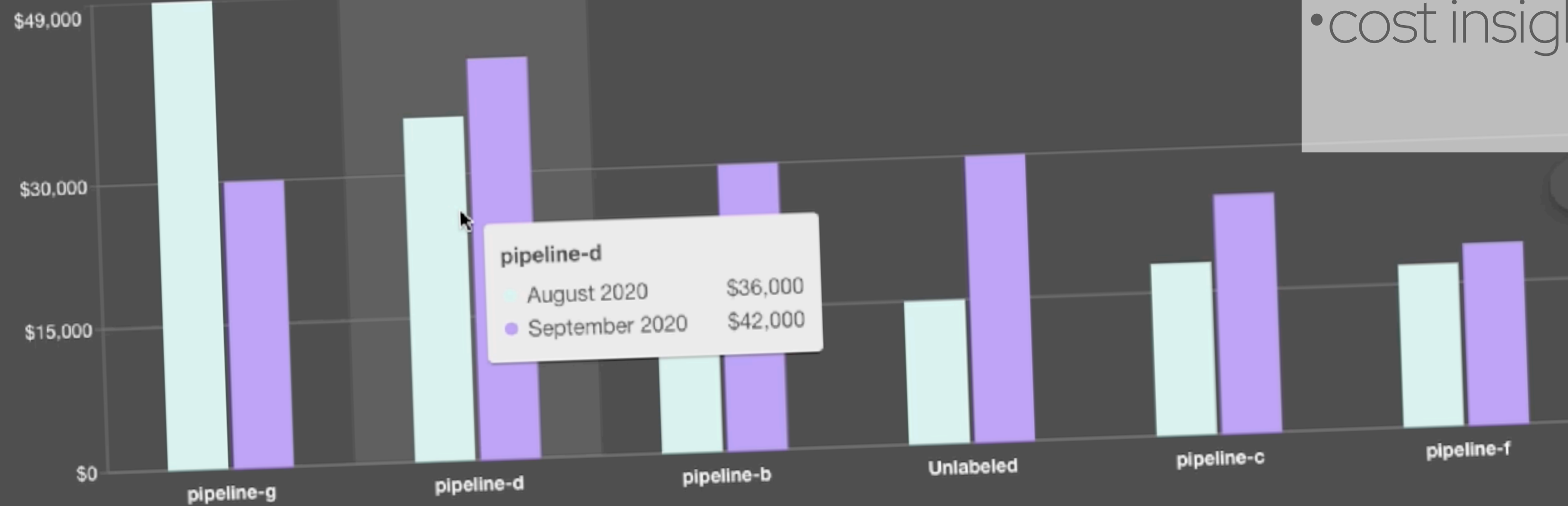
August 2020	\$36,000
September 2020	\$42,000

## Data Processing

9 entities, sorted by cost

August vs September ▾

● AUGUST 2020 ● SEPTEMBER 2020 COST GROWTH  
\$200,000 \$250,000 20% or ~3 engineers



backstage.io

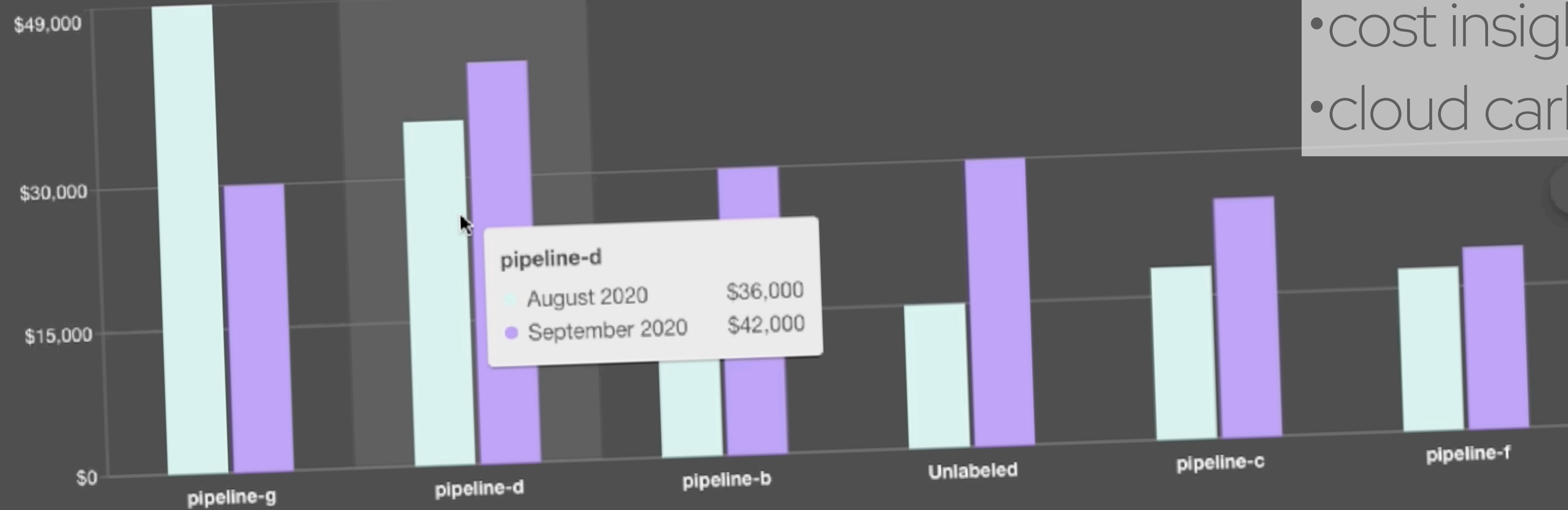
• cost insights plugin

## Data Processing

9 entities, sorted by cost

August vs September ▾

● AUGUST 2020 ● SEPTEMBER 2020 COST GROWTH  
\$200,000 \$250,000 20% or ~3 engineers



backstage.io

- cost insights plugin
- cloud carbon footprint plugin



# AIOps

- Densify
- Granulate
- Turbonomic Application Resource Management
- TSO Logic
- etc

21%

improvement from installing Turbonomic  
in IBM CIO office

traffic monitoring

but.


knowing is only half the battle.

the ikea effect

the ikea effect

labour

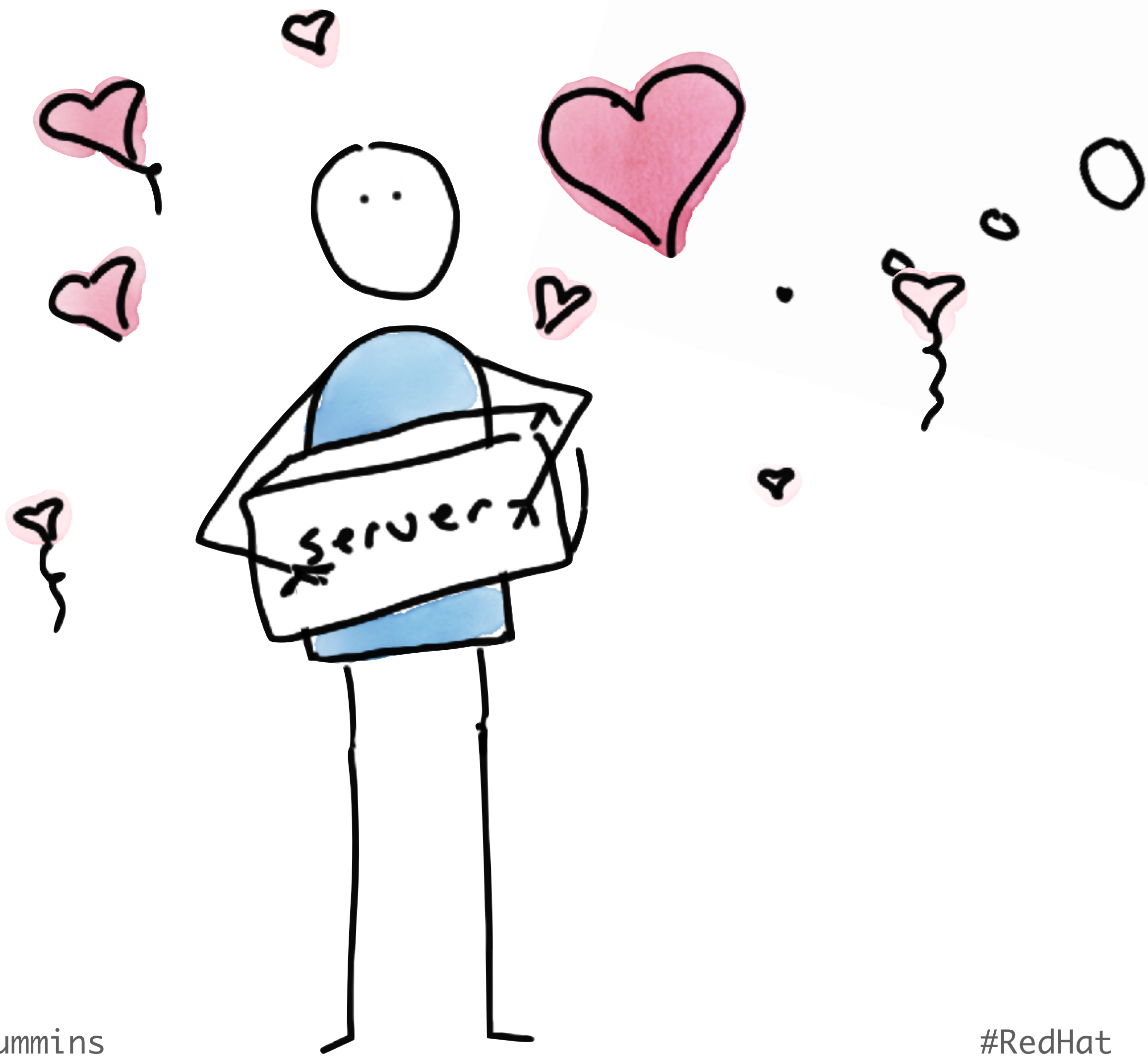
the ikea effect

labour 

the ikea effect

labour → love



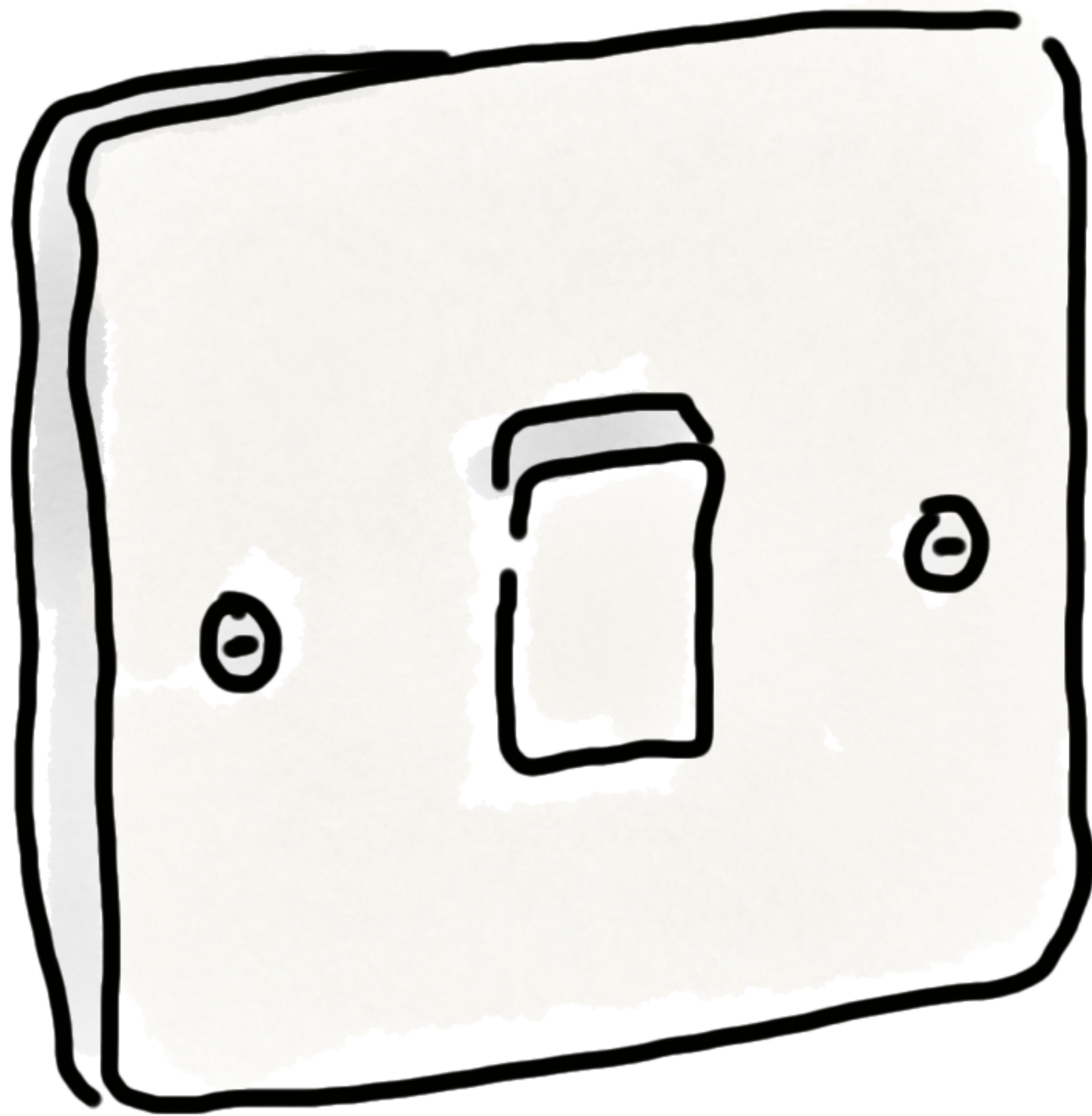


shut it down?  
but ... what if I  
**need** this  
cluster later?

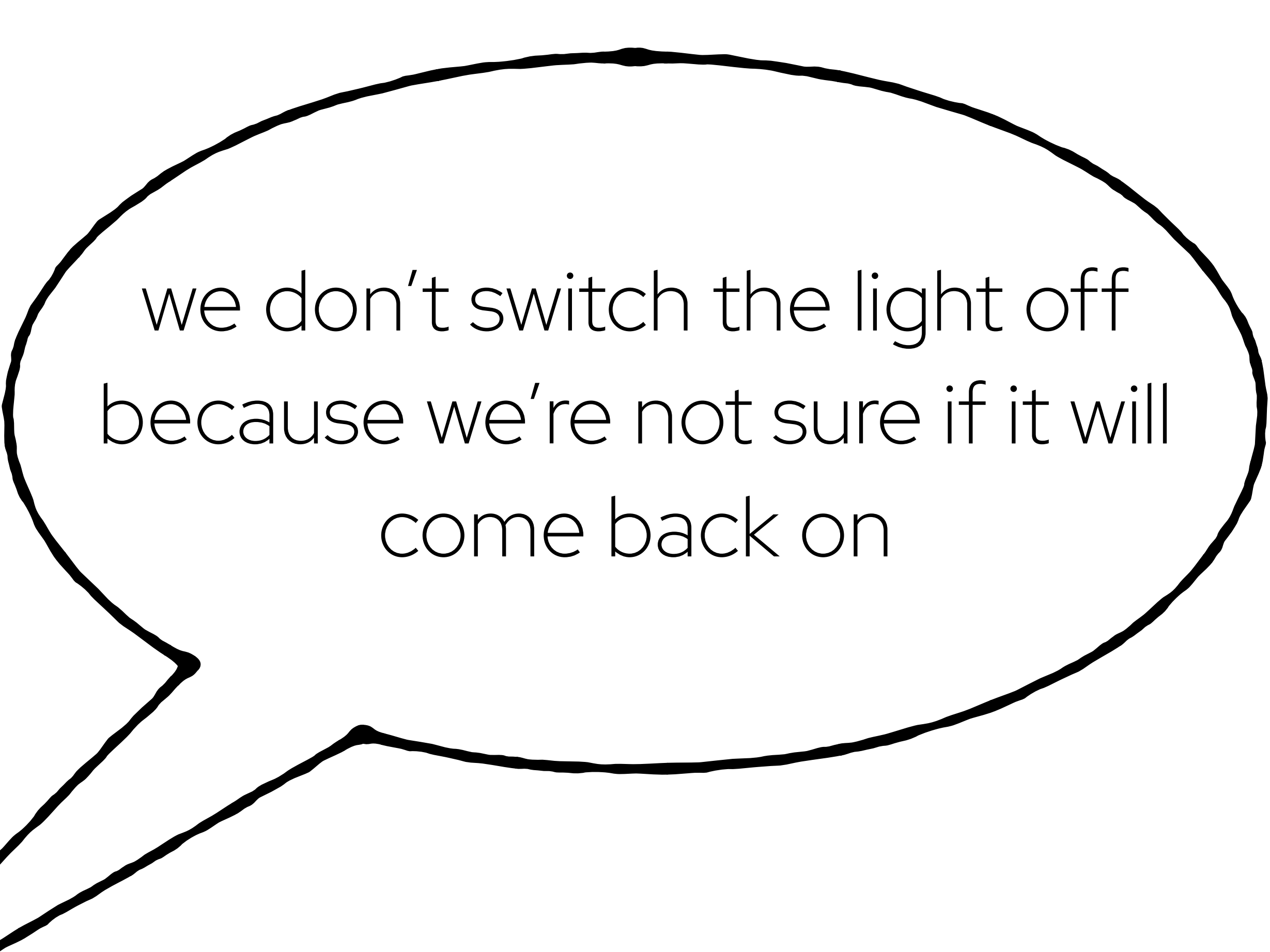
elasticity

native quarkus starts  
faster than a light bulb

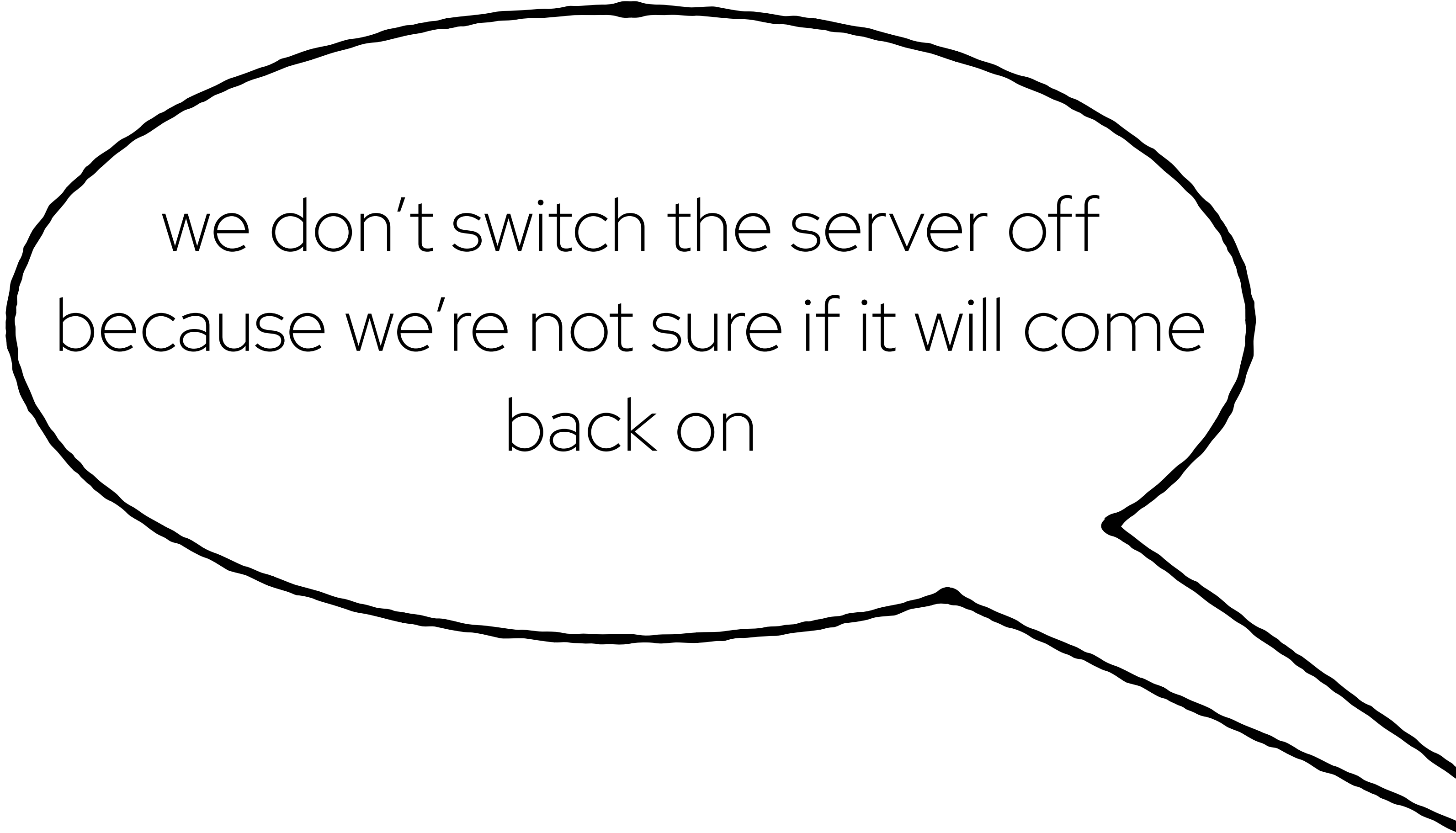




ultimate elasticity

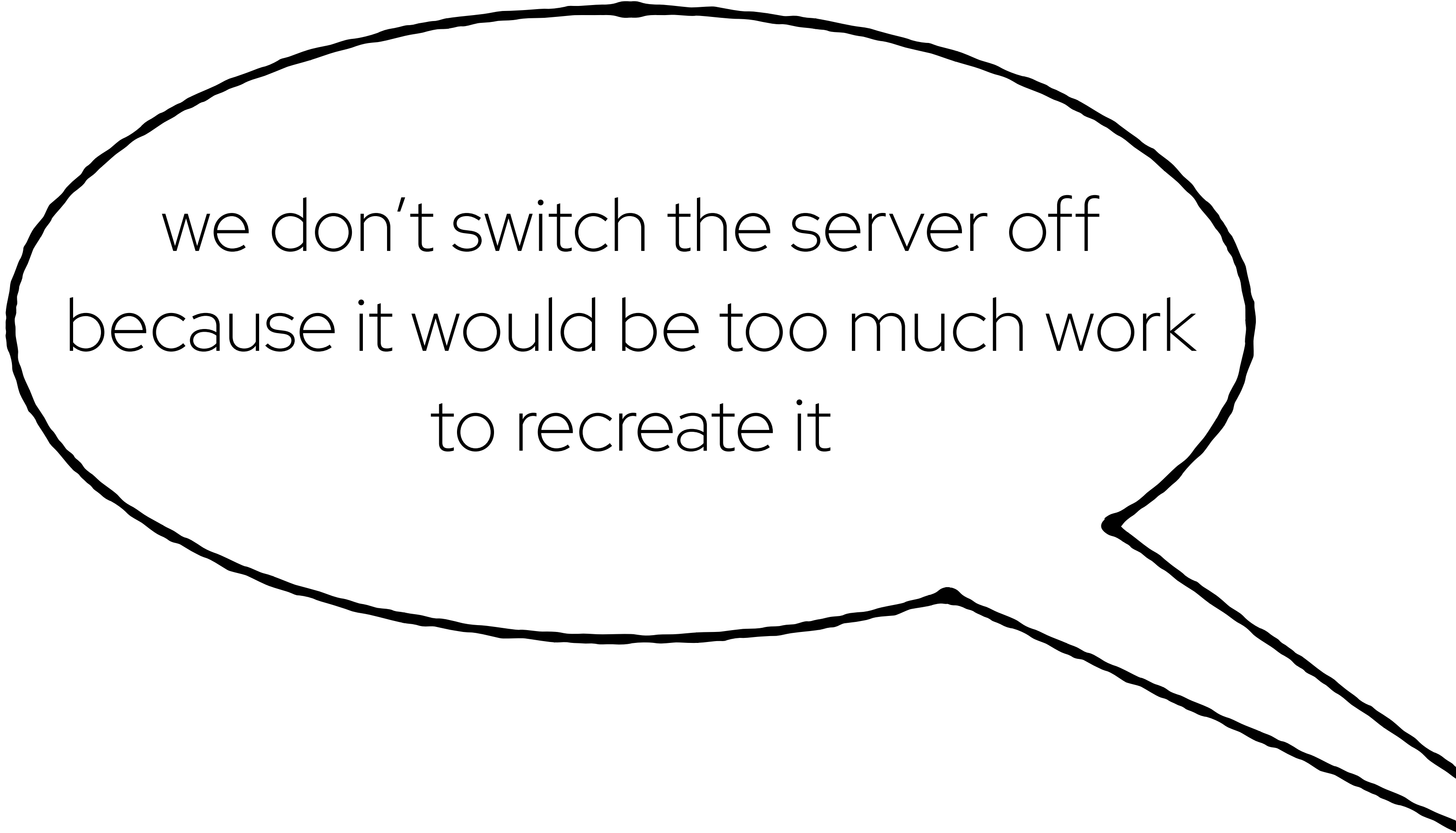
A hand-drawn speech bubble with a thick black outline. The bubble is roughly oval-shaped with a small tail pointing towards the bottom-left corner. Inside the bubble, the text is centered and reads: "we don't switch the light off because we're not sure if it will come back on".

we don't switch the light off  
because we're not sure if it will  
come back on



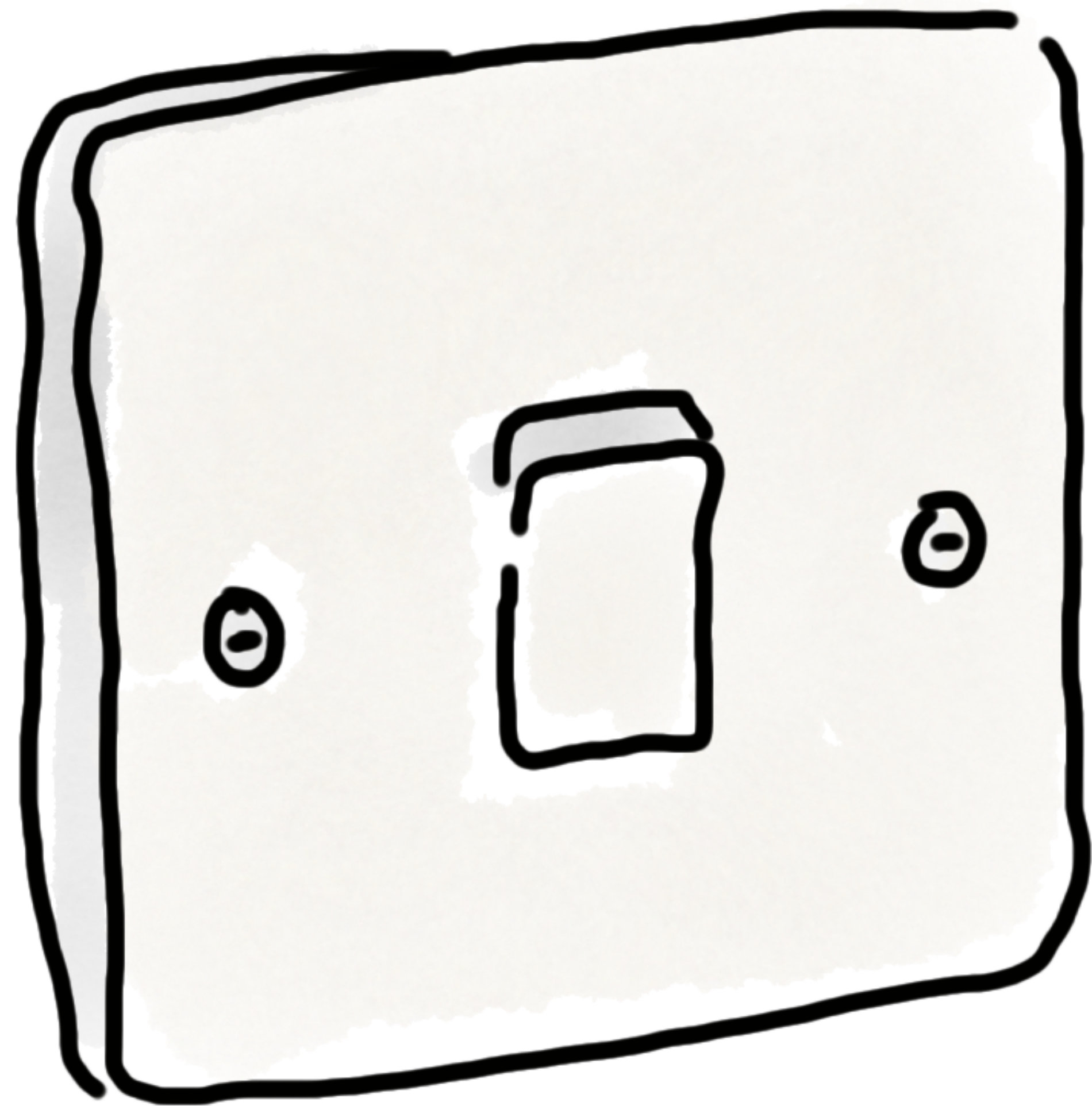
we don't switch the server off  
because we're not sure if it will come  
back on

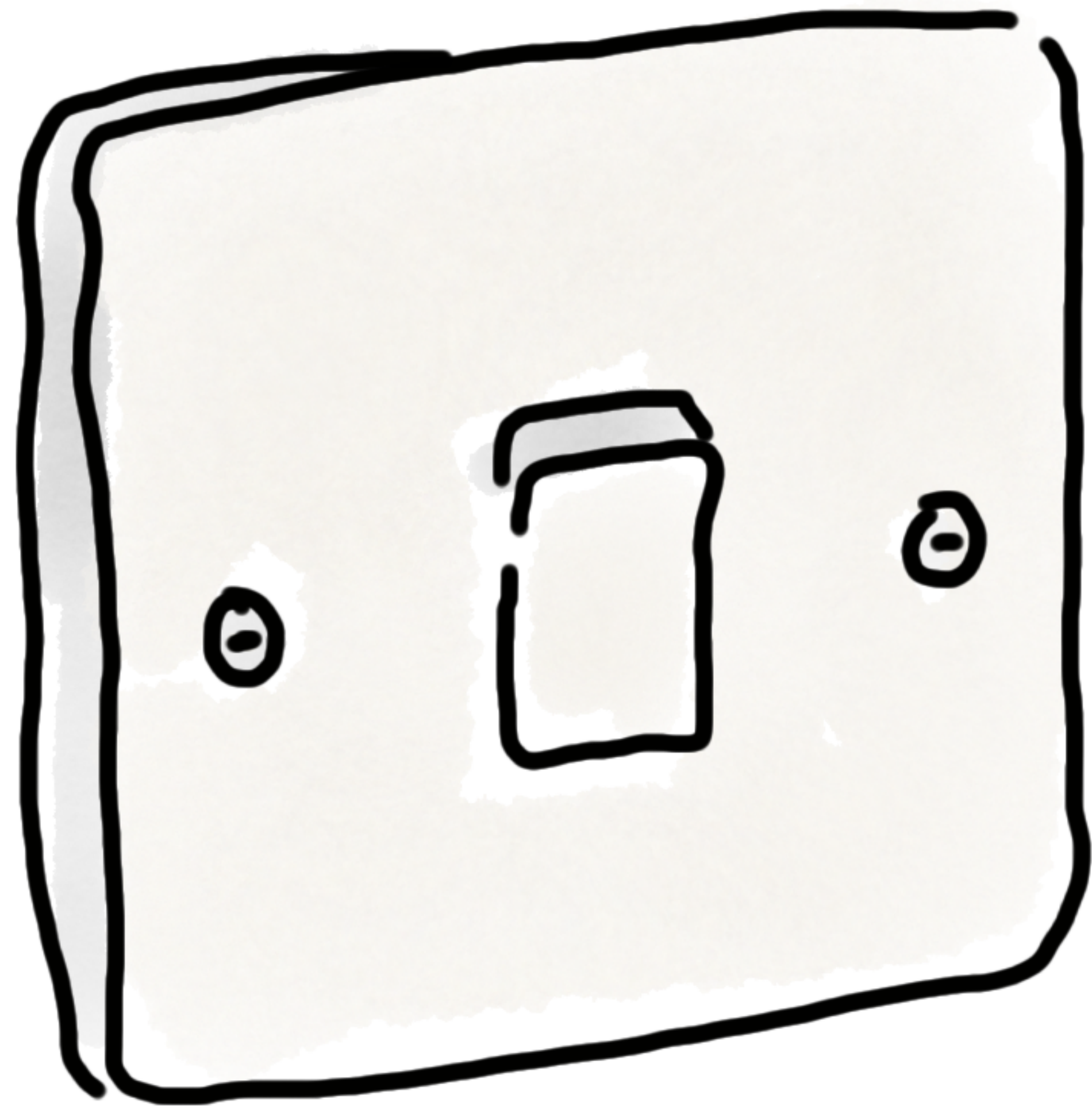
happens **all** the time



we don't switch the server off  
because it would be too much work  
to recreate it

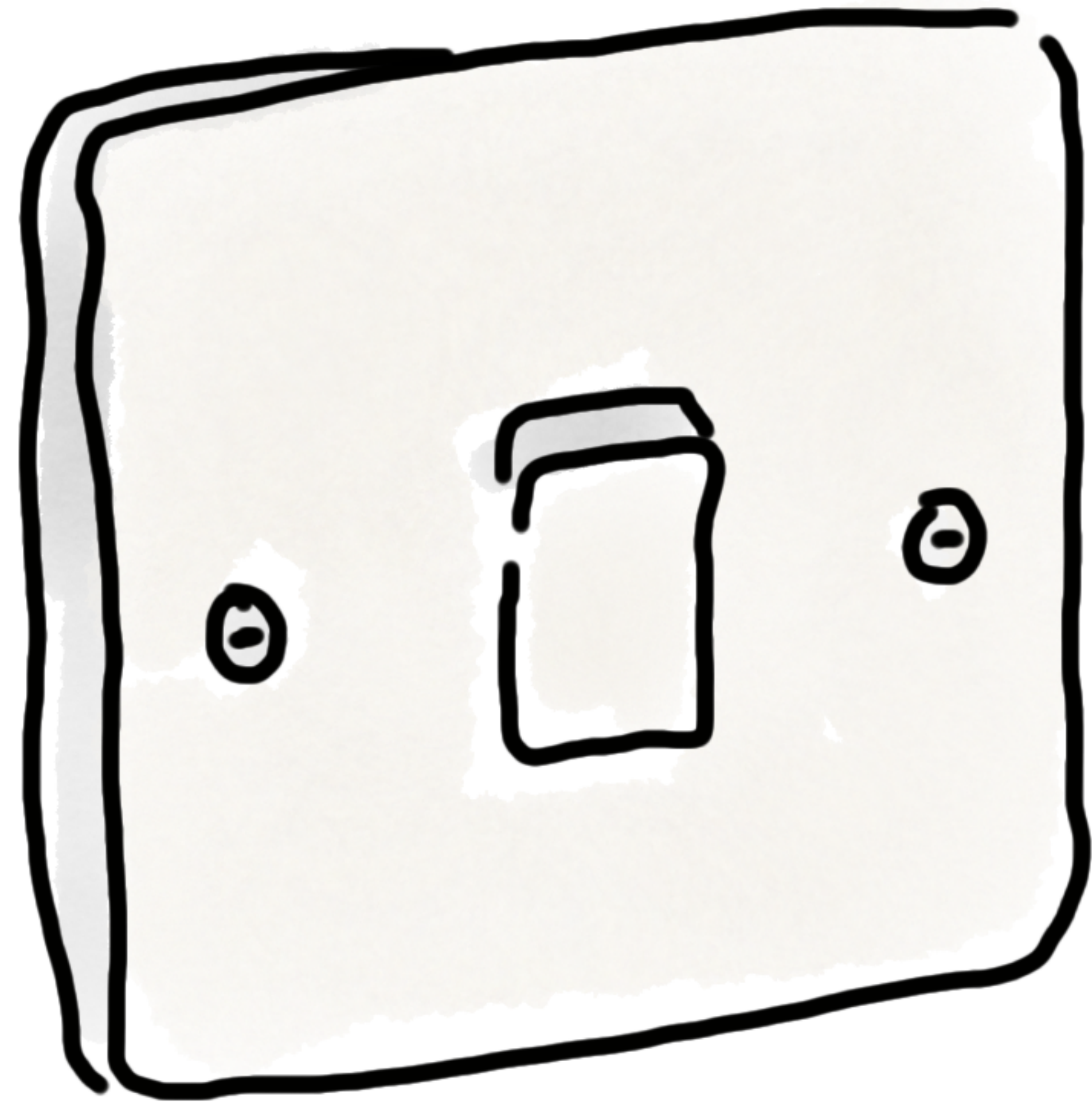
happens **all** the time





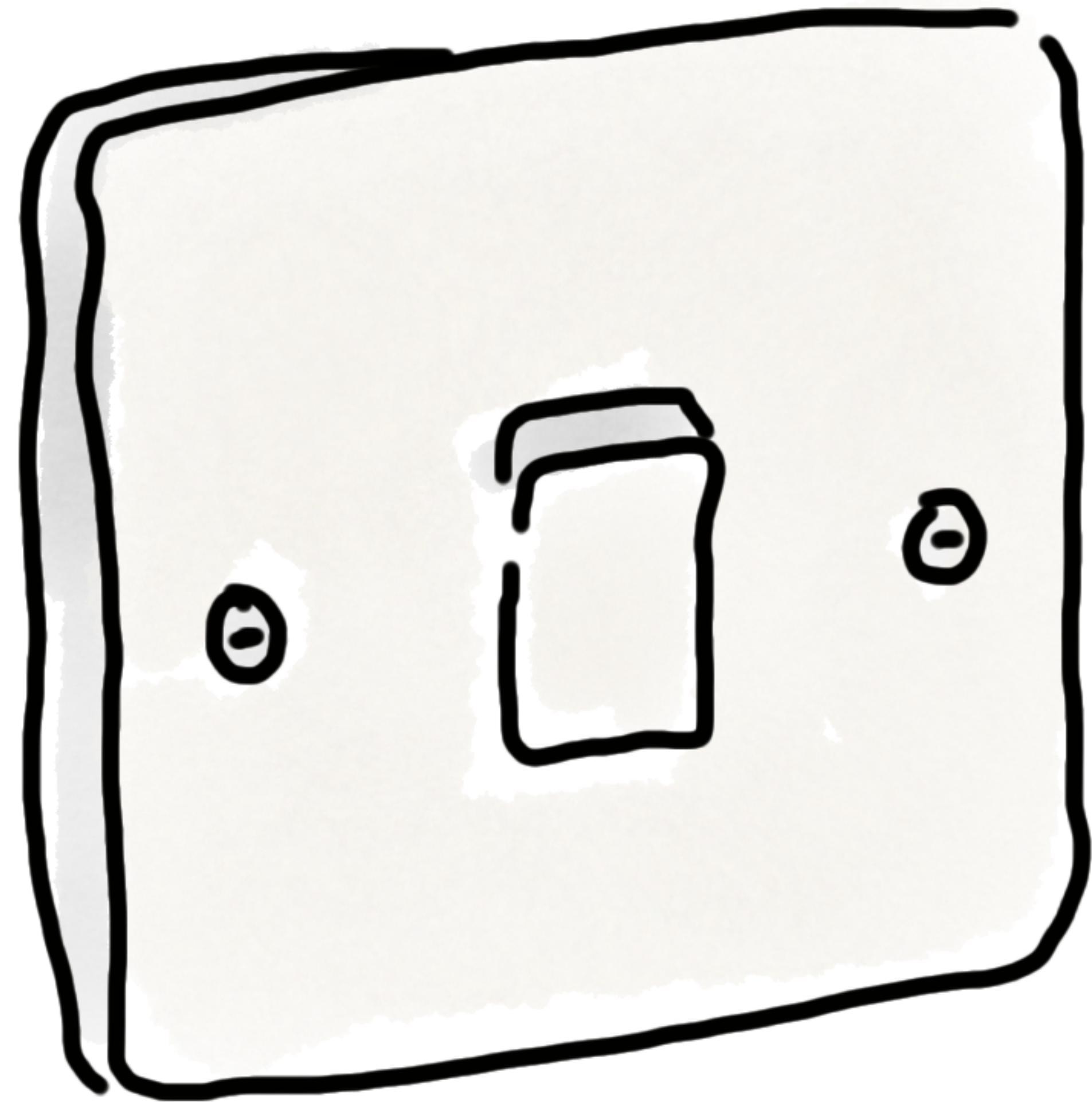


turning it off and on again must



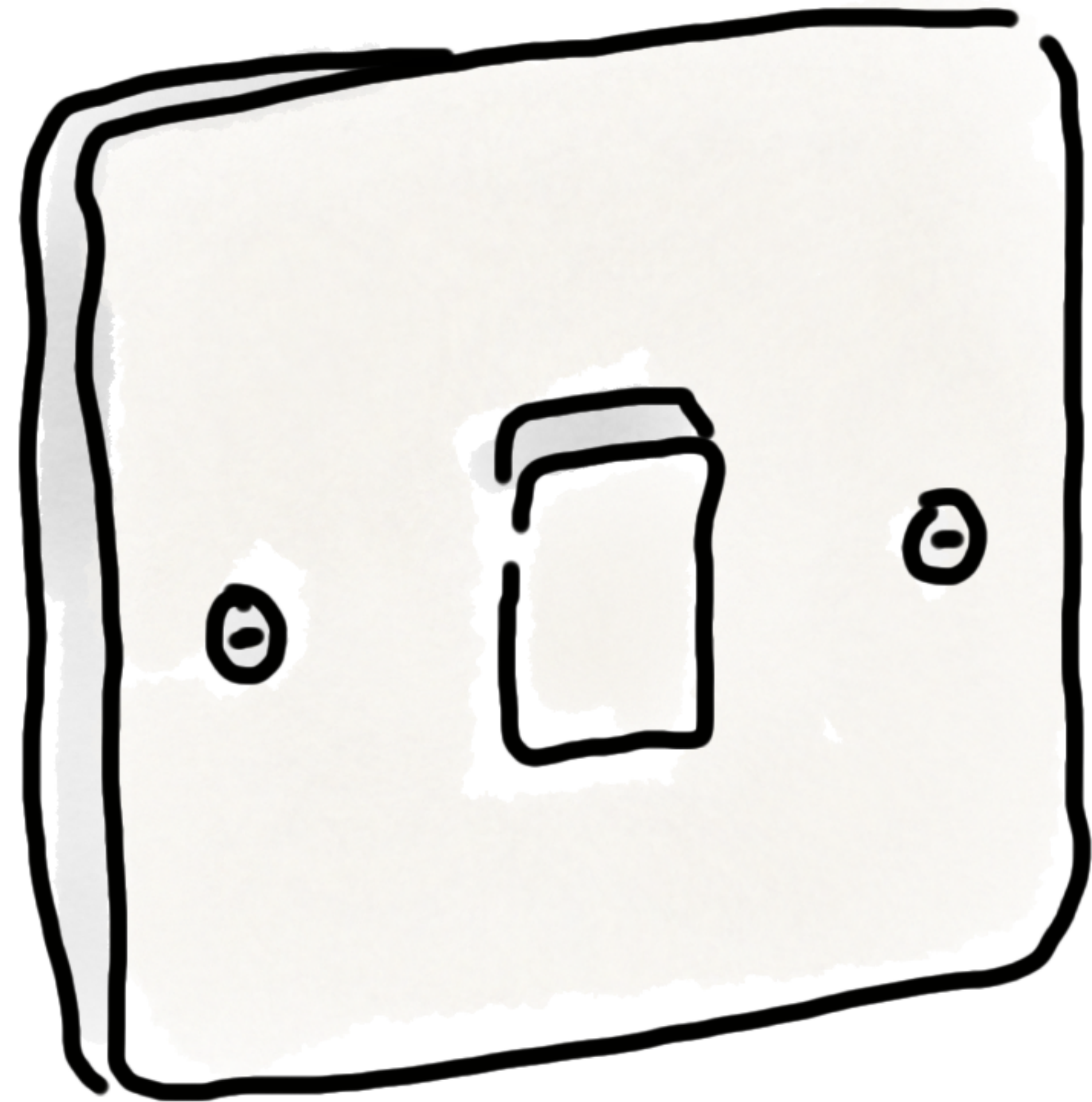
turning it off and on again must

- be **fast**



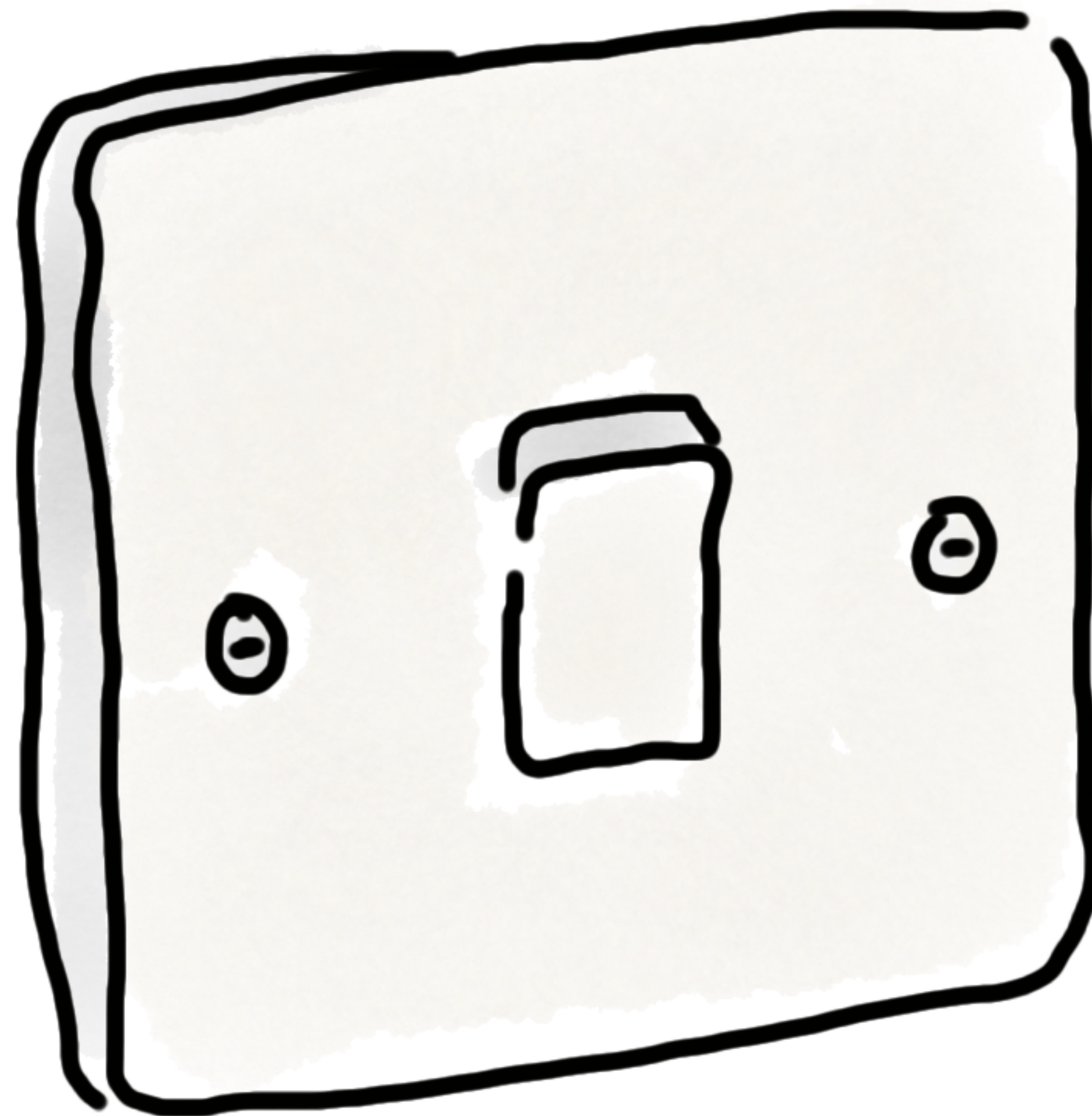
turning it off and on again must

- be **fast**
- actually **work**



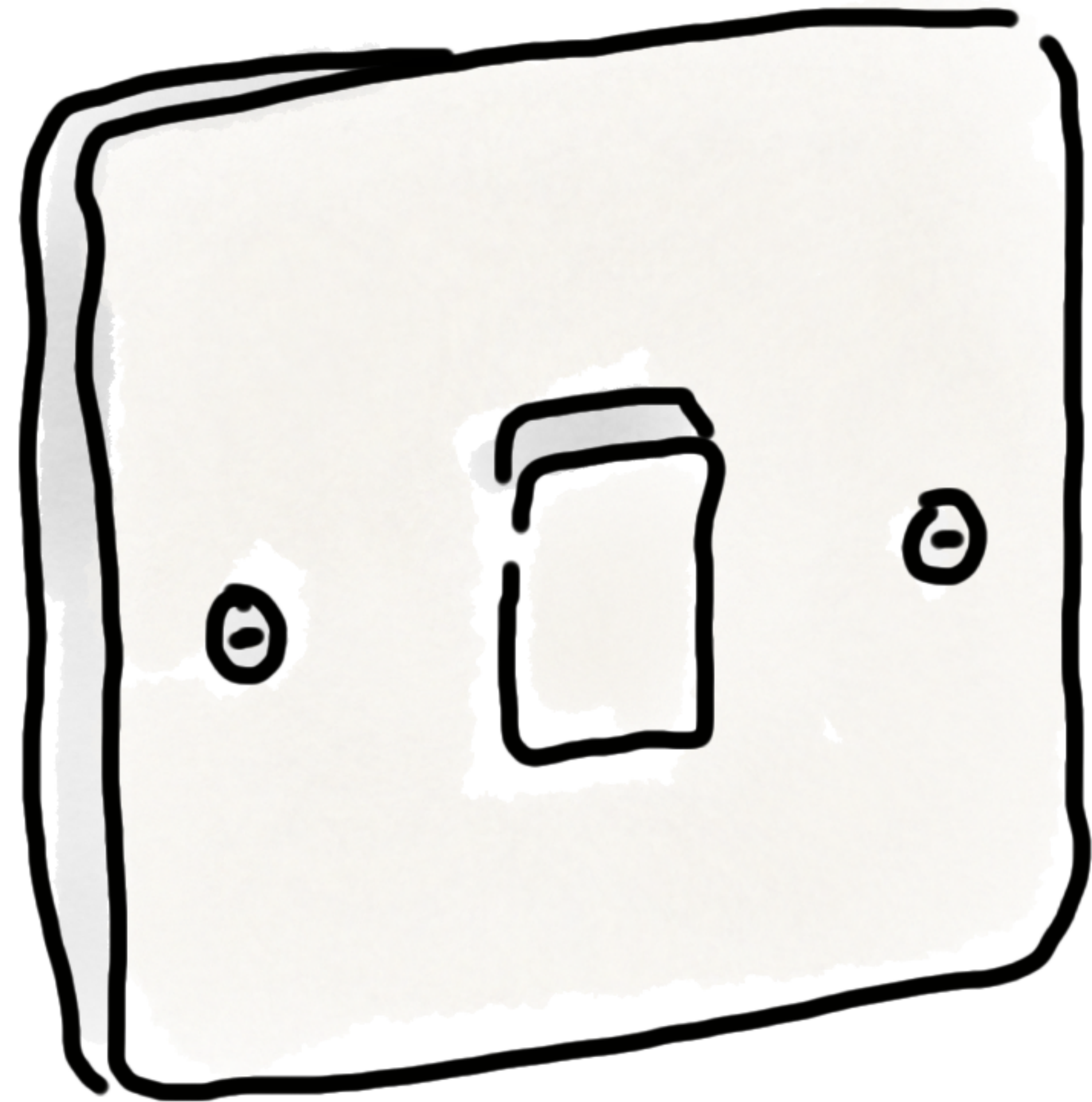
turning it off and on again must

- be **fast**
- actually **work**
  - idempotency

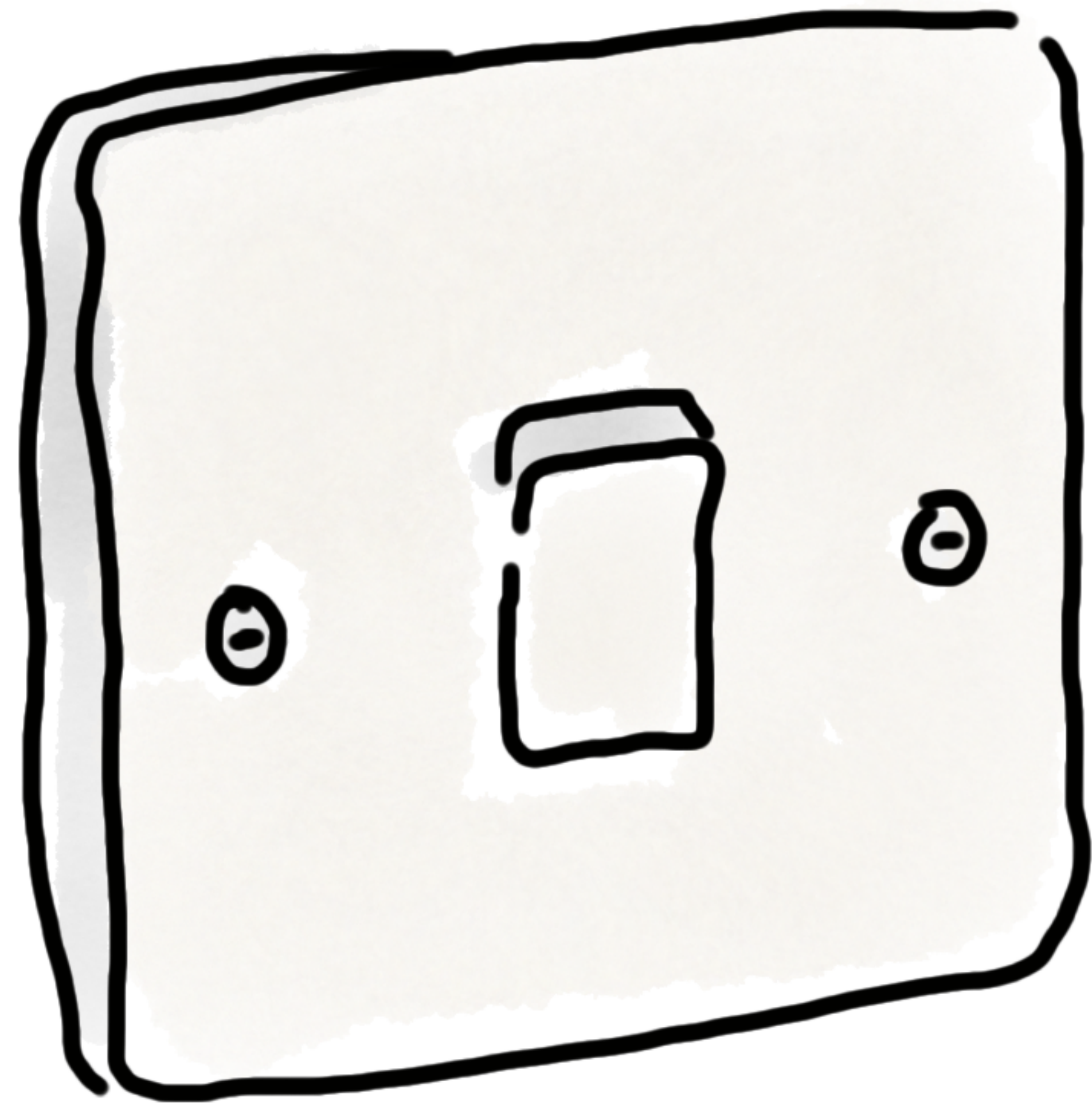


turning it off and on again must

- be **fast**
- actually **work**
  - idempotency
  - resiliency

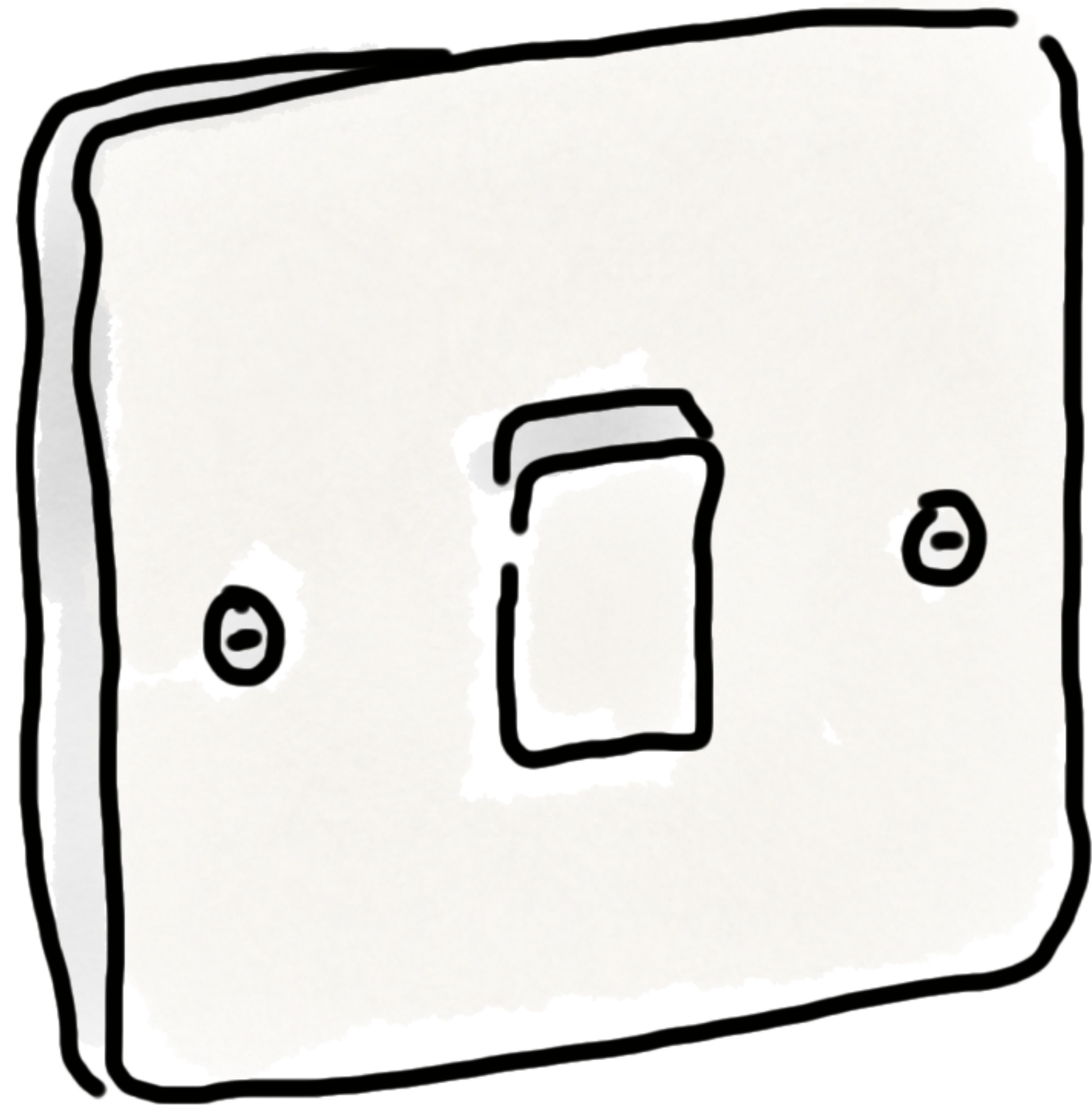


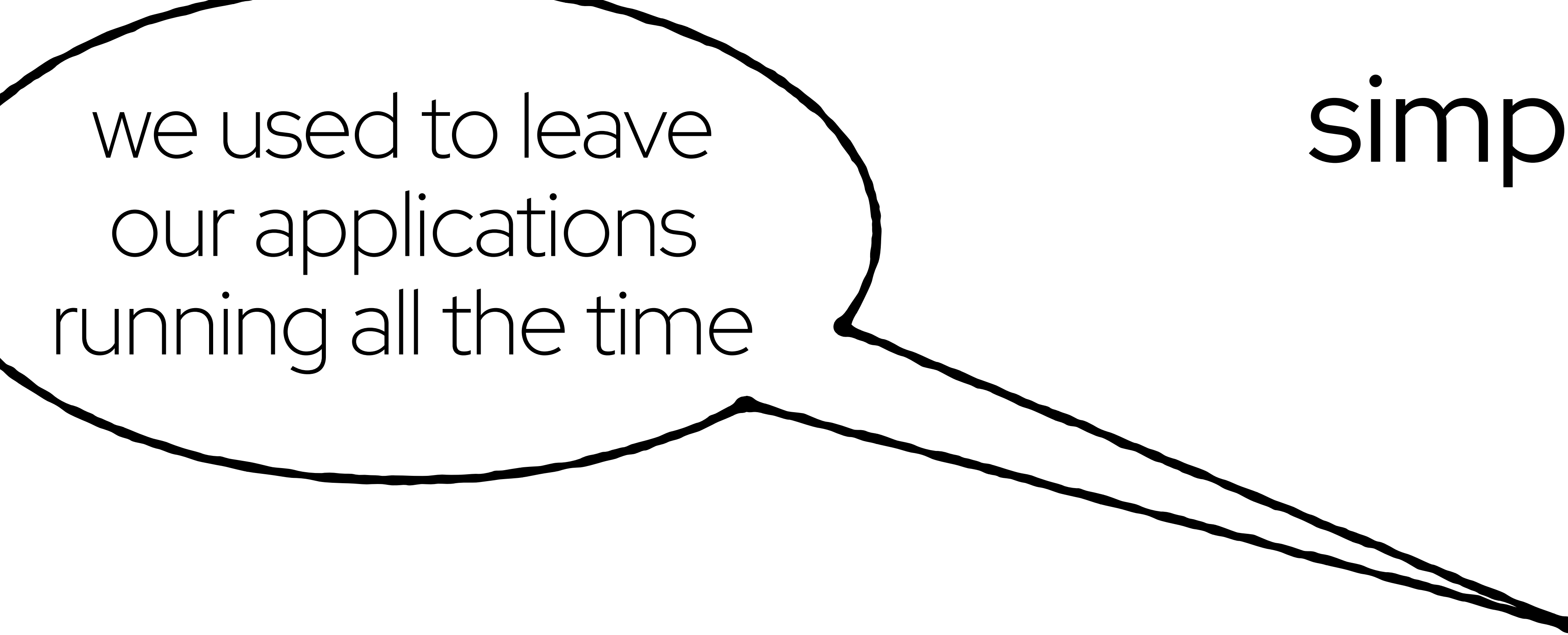
making turning servers off as safe and easy as turning lights off



# LightSwitchOps

making turning servers off as safe and easy as turning lights off





we used to leave  
our applications  
running all the time

# simple scripts

@darkandnerdy, Chicago DevOpsDays



# simple scripts

we used to leave  
our applications  
running all the time

when we  
scripted turning  
them off at night,  
we reduced our  
cloud bill by  
**30%**

@darkandnerdy, Chicago DevOpsDays

@holly\_cummins

#RedHat

# GitOps

# GitOps

(infrastructure as code)



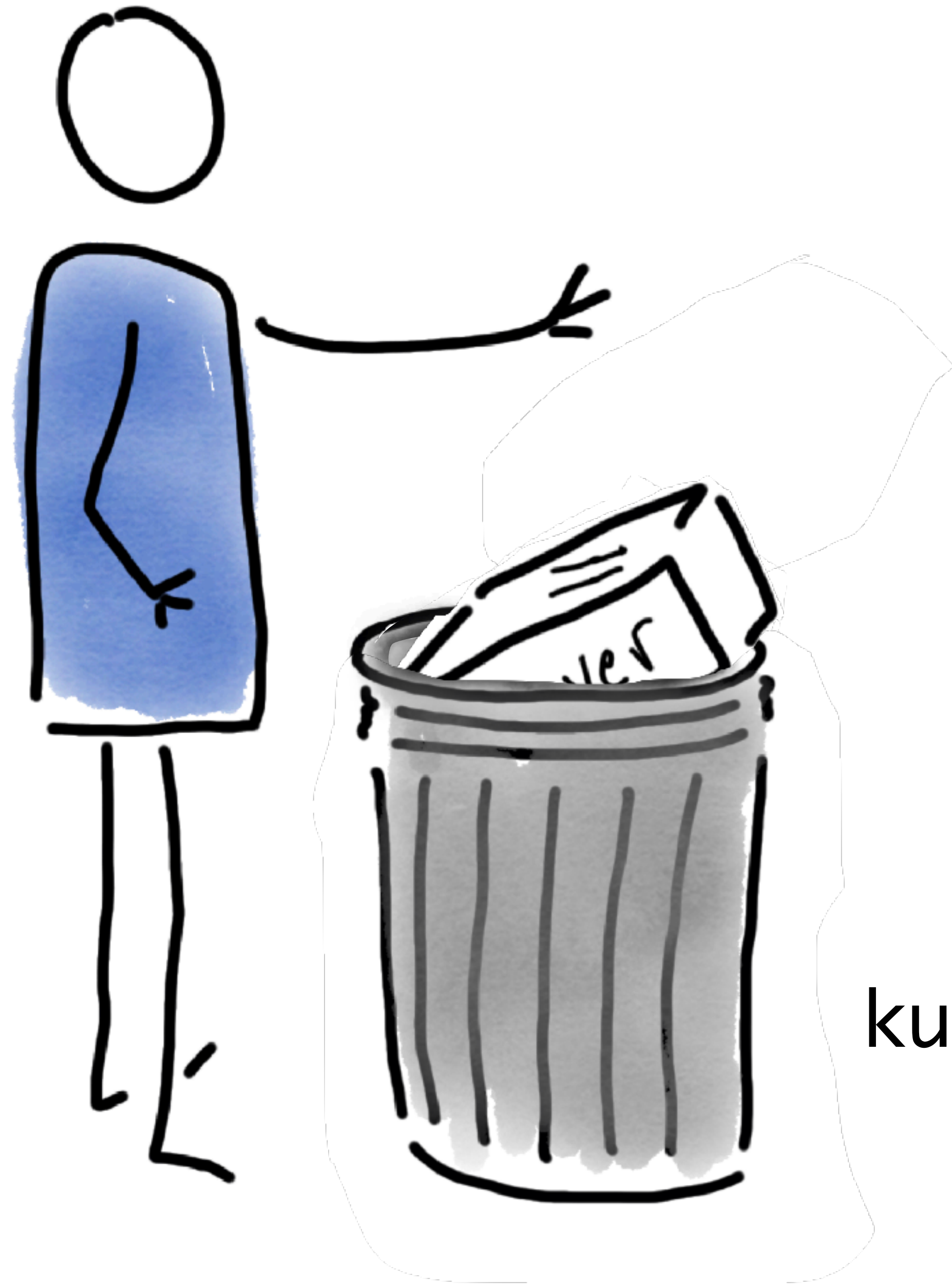


spin it down



spin it down  
spin it up

```
kubectl apply -f all-my-cluster/
```



spin it down  
spin it up

```
kubectl apply -f all-my-cluster/
```

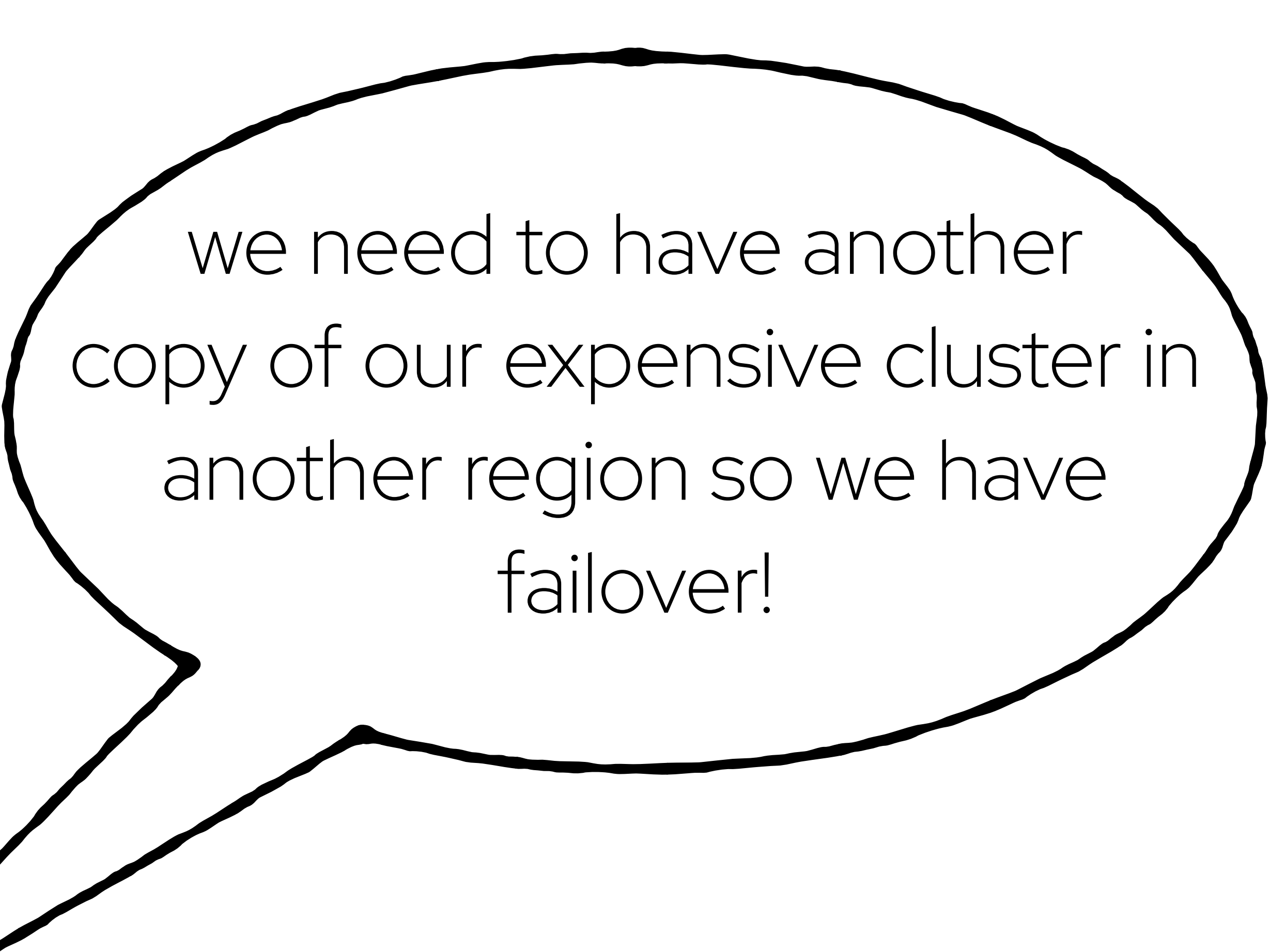




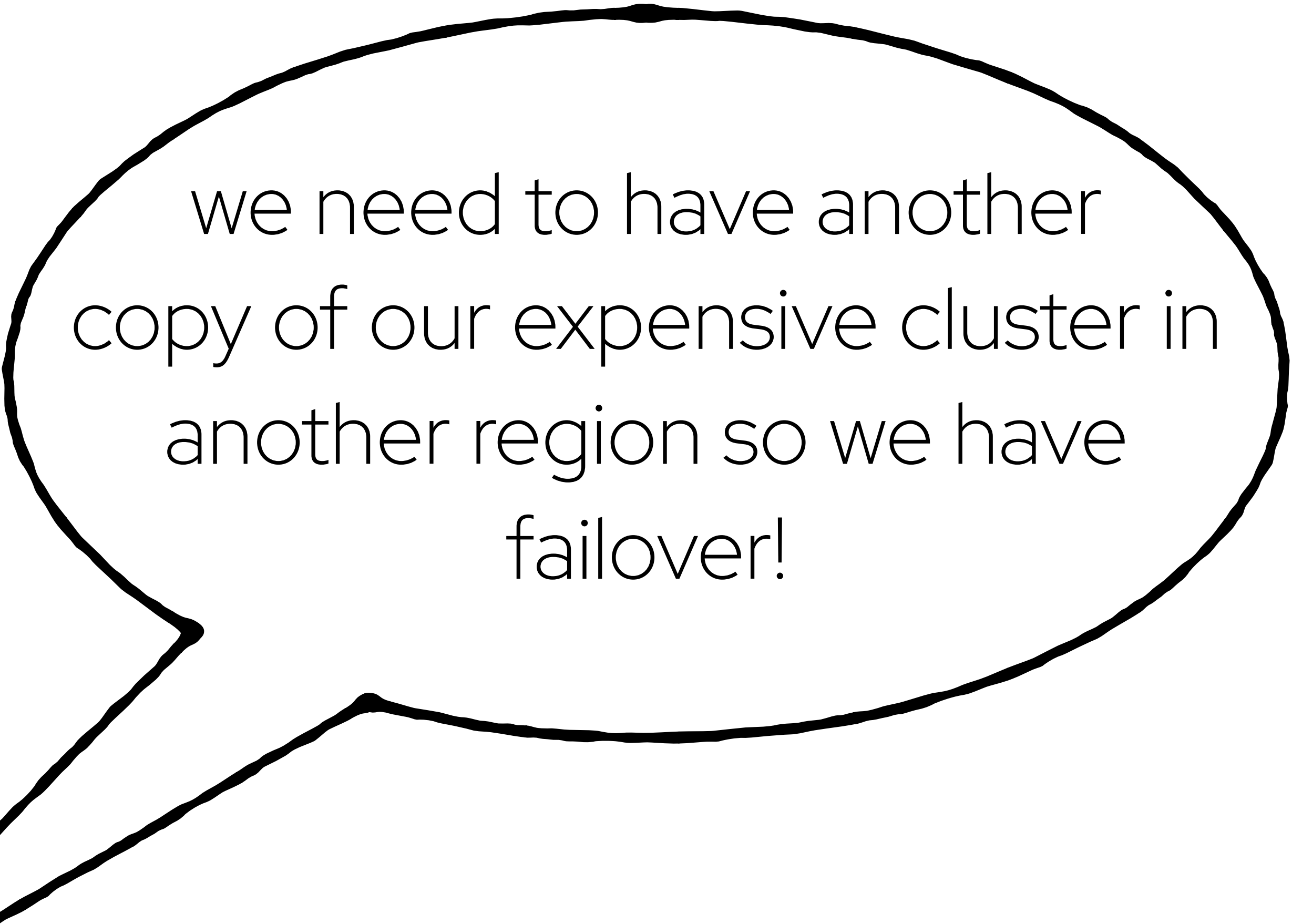
spin it down  
spin it up

ansible-playbook stuff.yml

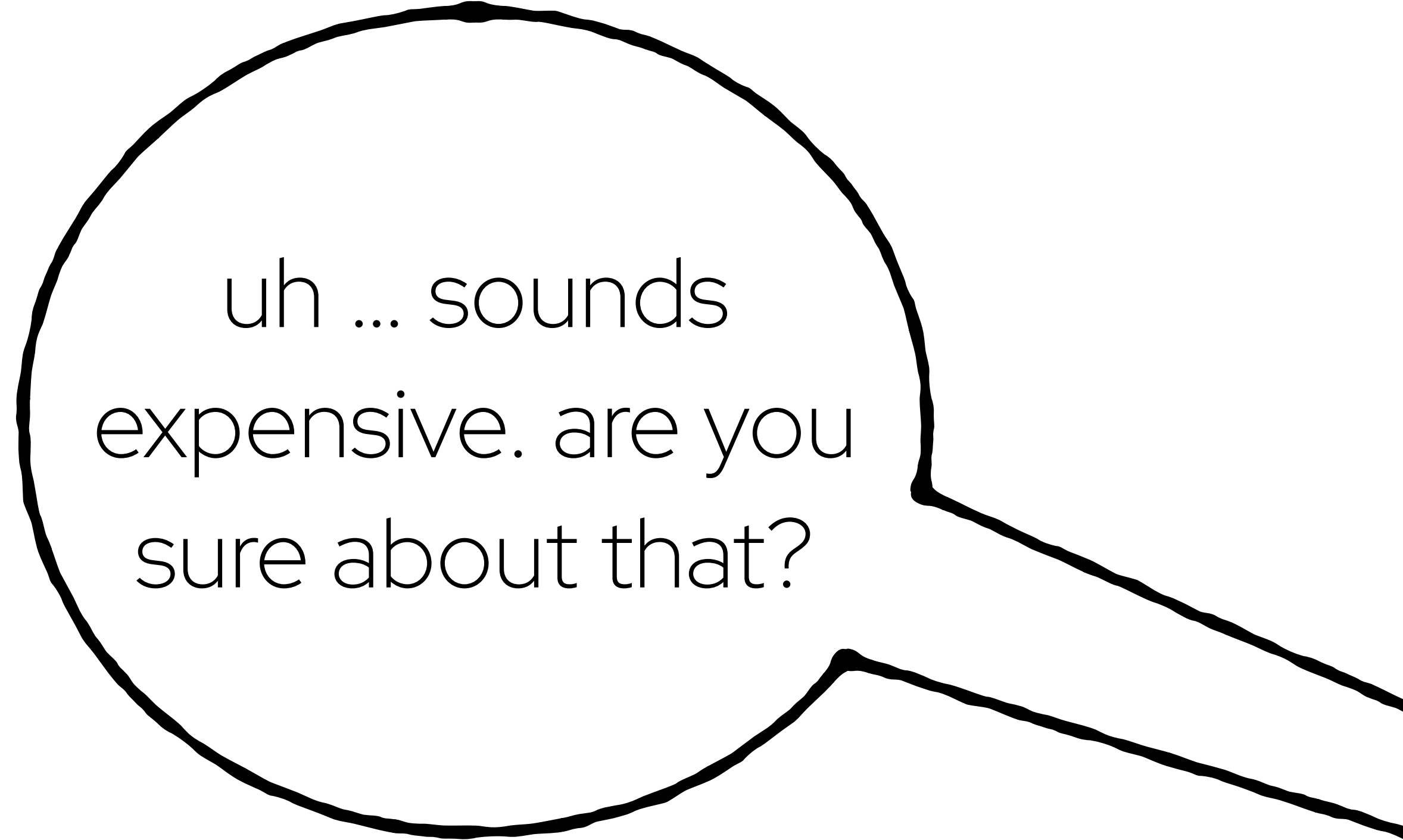
reducing snowflakes  
reduces redundancy

A hand-drawn speech bubble with a thick black outline. The bubble is roughly oval-shaped with a small tail pointing towards the bottom-left corner. Inside the bubble, the text is centered and reads: "we need to have another copy of our expensive cluster in another region so we have failover!".

we need to have another  
copy of our expensive cluster in  
another region so we have  
failover!



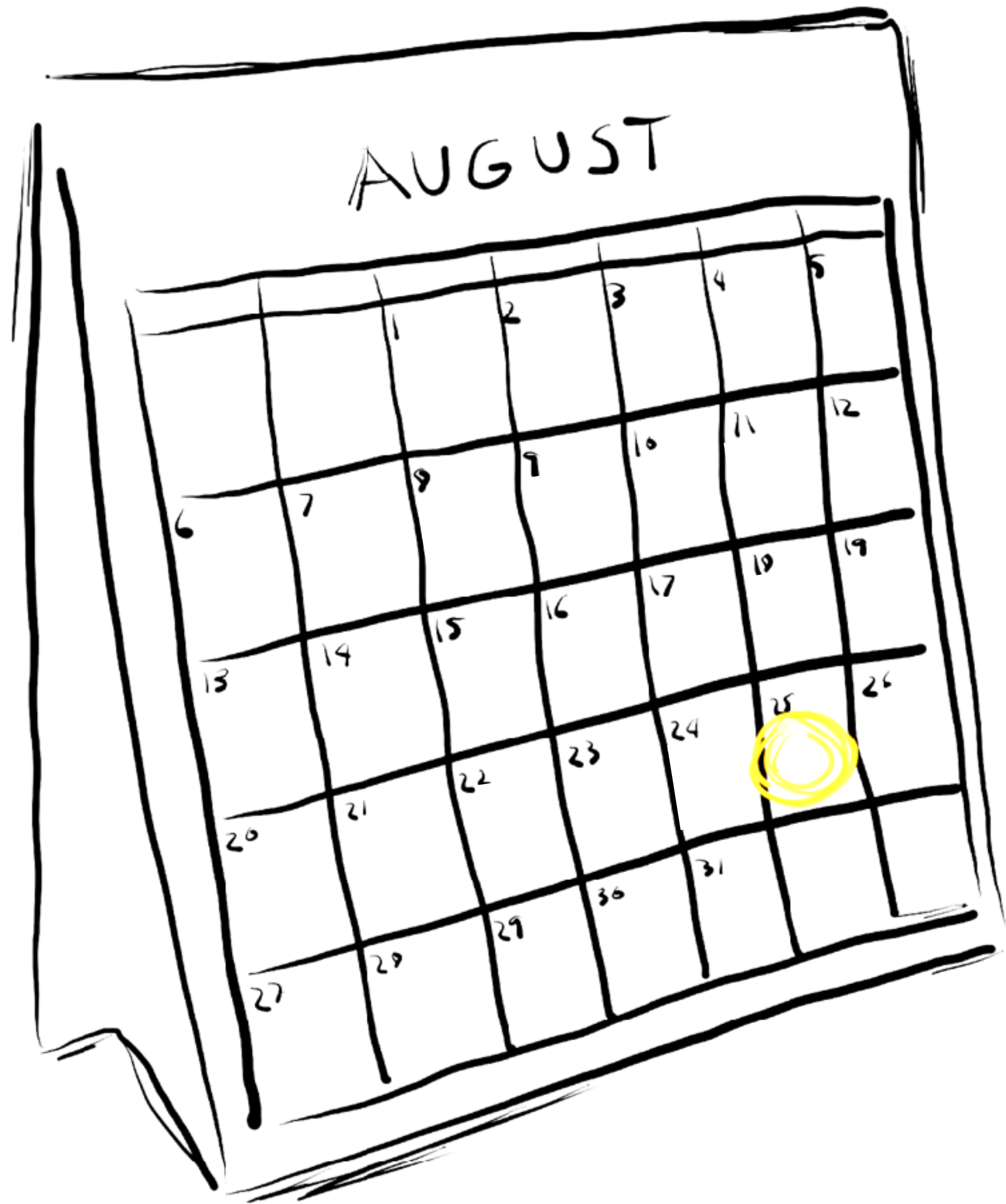
we need to have another copy of our expensive cluster in another region so we have failover!



uh ... sounds expensive. are you sure about that?

rapid recovery does **not**  
require redundant servers

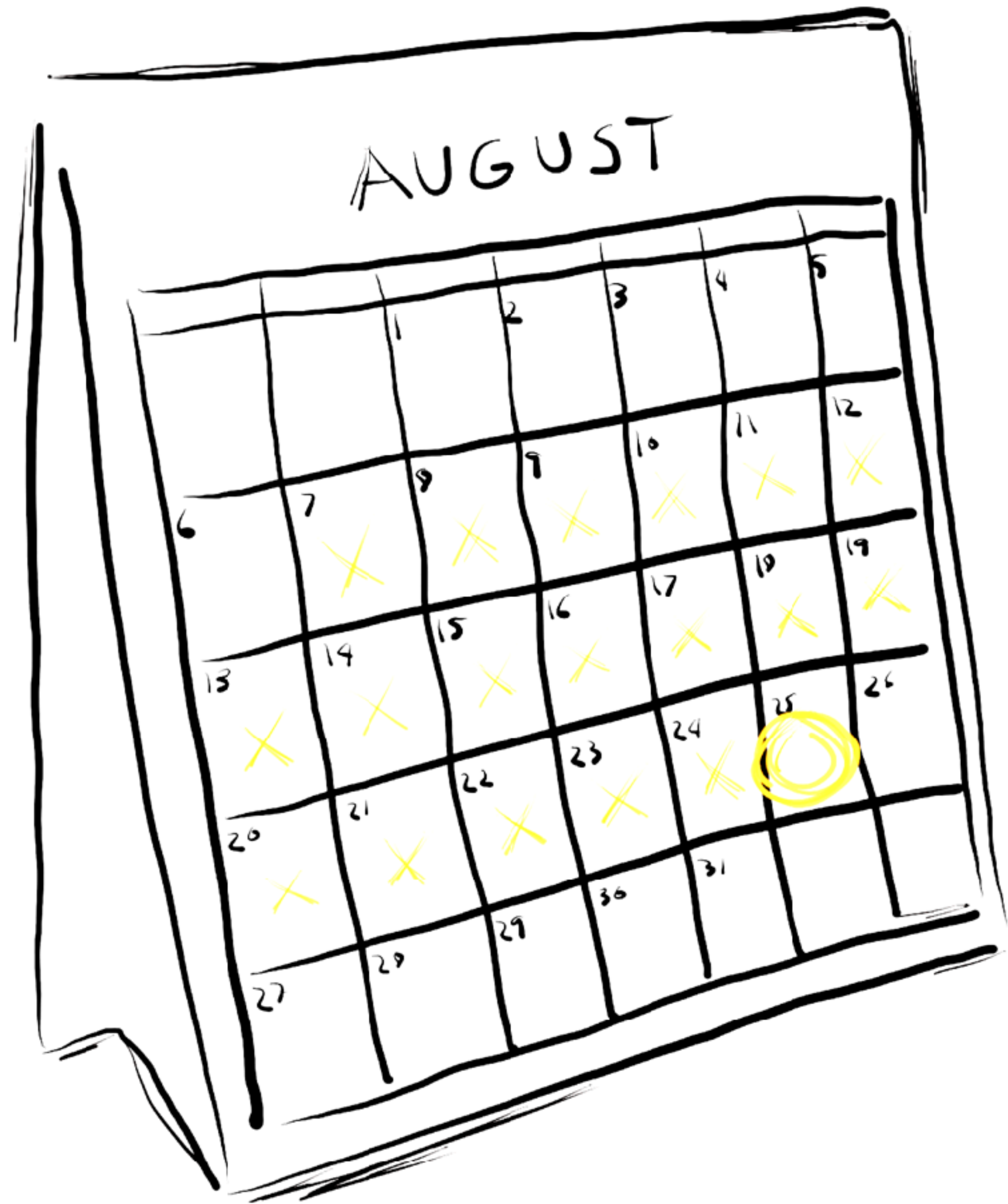
zombie reduction does  
not need to be fancy



large bank, 2013

50%

reduction in CPUs with a  
lease system



large bank, 2013

# 50%

reduction in CPUs with a  
lease system



things that (maybe) don't help

things that (maybe) don't help

cloud



“out of sight, out of mind”



**Corey Quinn** @QuinnyPig · Jul 29, 2020



Replying to [@QuinnyPig](#)

The beauty of cloud is in its elasticity. It lets you scale up to meet traffic demands, and then when that traffic wanes you can keep your scaled up environment running in perpetuity to help send some engineers' kids to college.



things that (maybe) don't help

# virtualisation

2019 survey

30%

of **virtual** servers doing  
no useful work

things that (maybe) don't help

# virtualisation

2019 survey

30%

of **virtual** servers doing  
no useful work

50%

of virtual servers active  
less than 5% of the time

you still need to remember to  
turn the virtual machine off

what about serverless?

modernising to serverless is a big lift

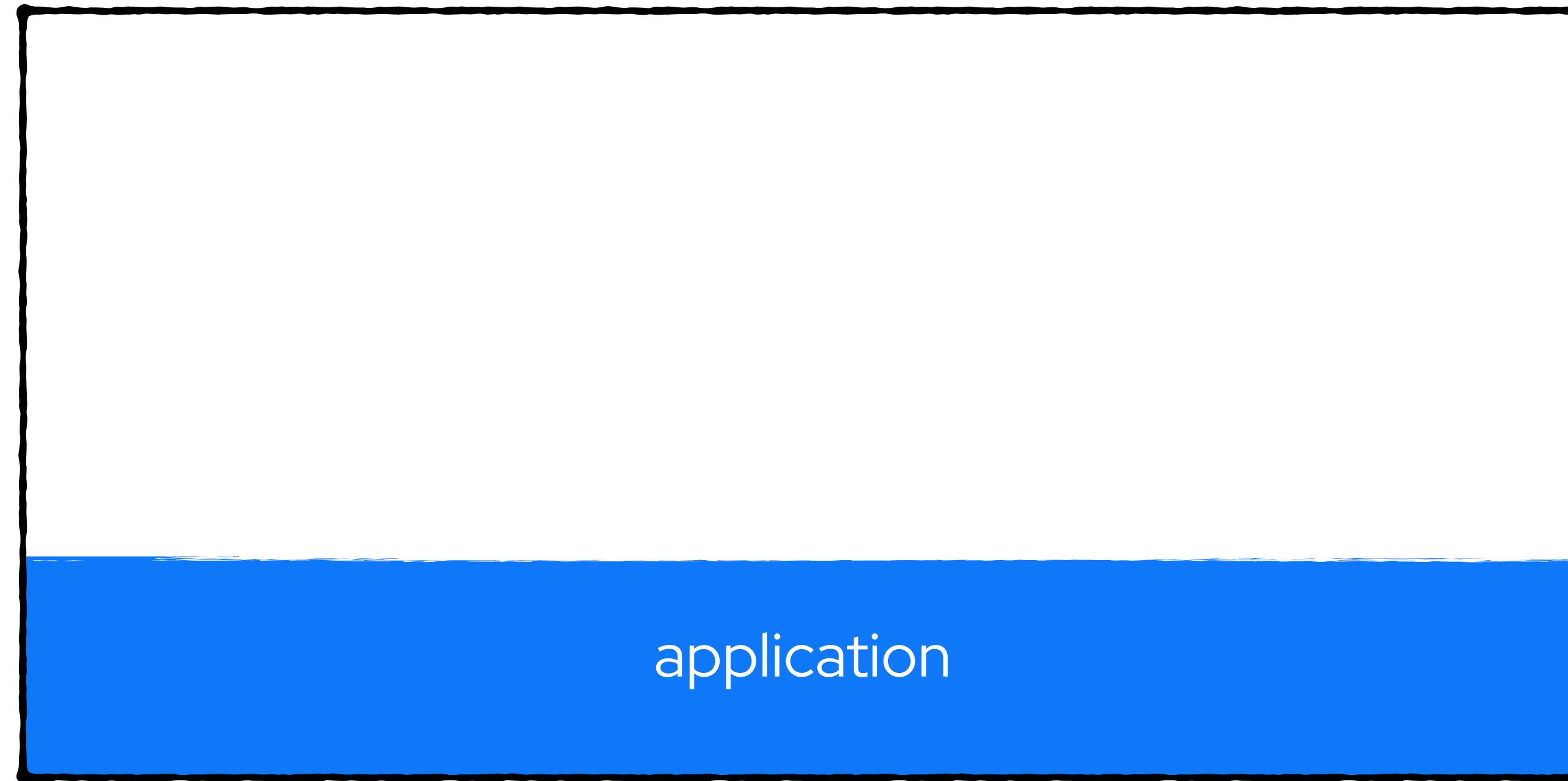


may not suit latency-sensitive workloads

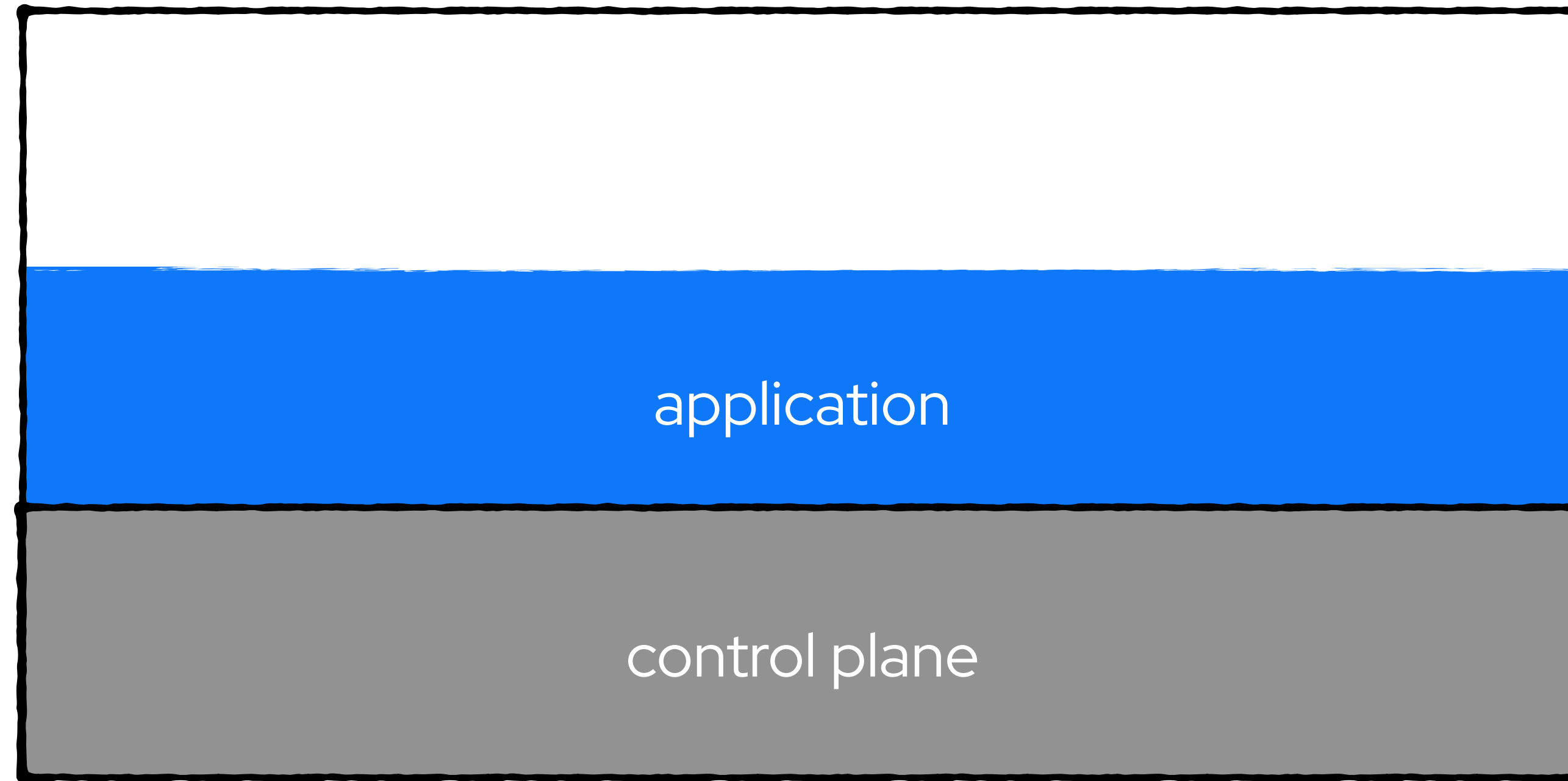
“we solve the cold-start problem by ...

... keeping an instance running but not billing you”

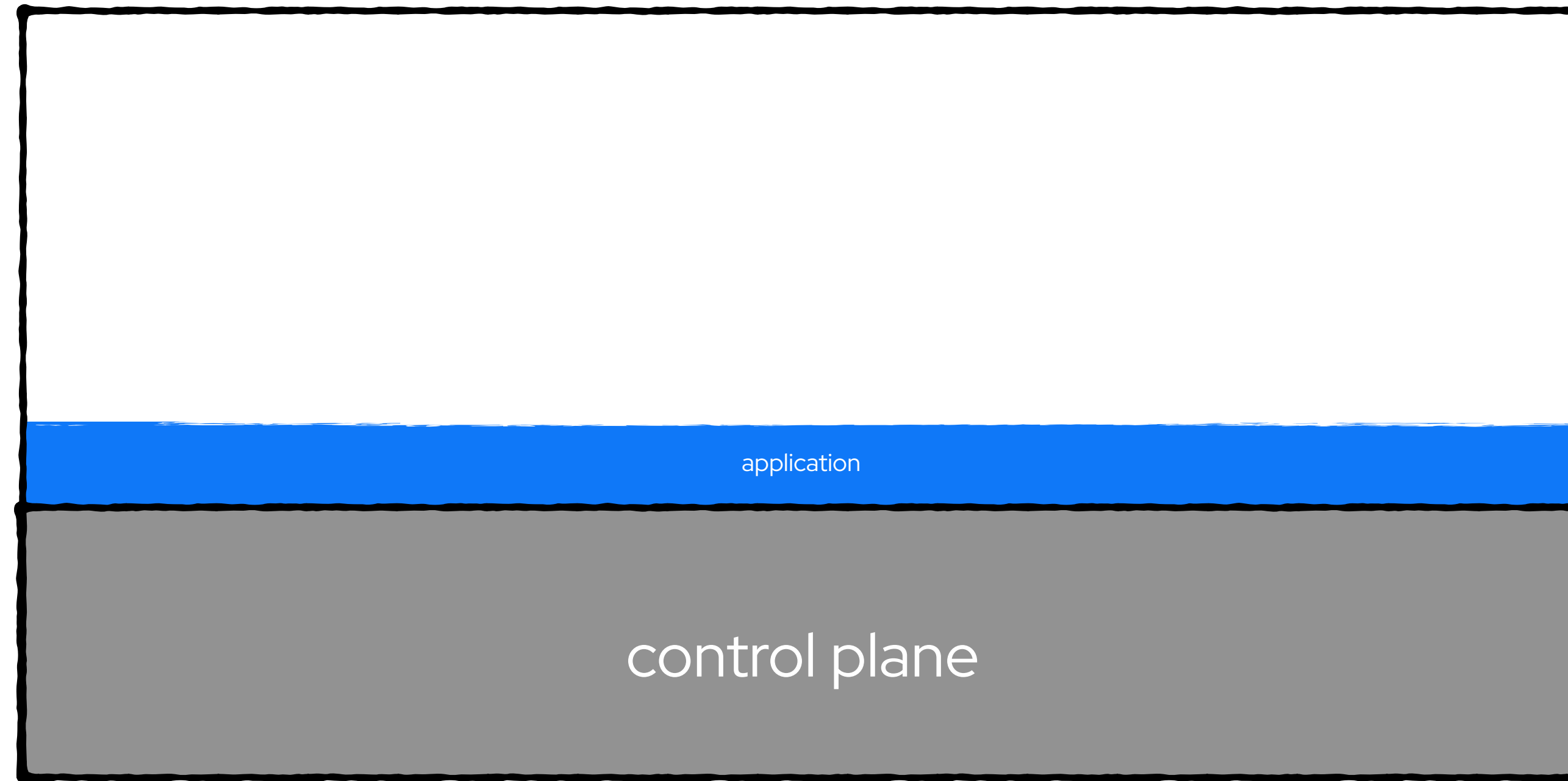
serverless systems may have high overheads



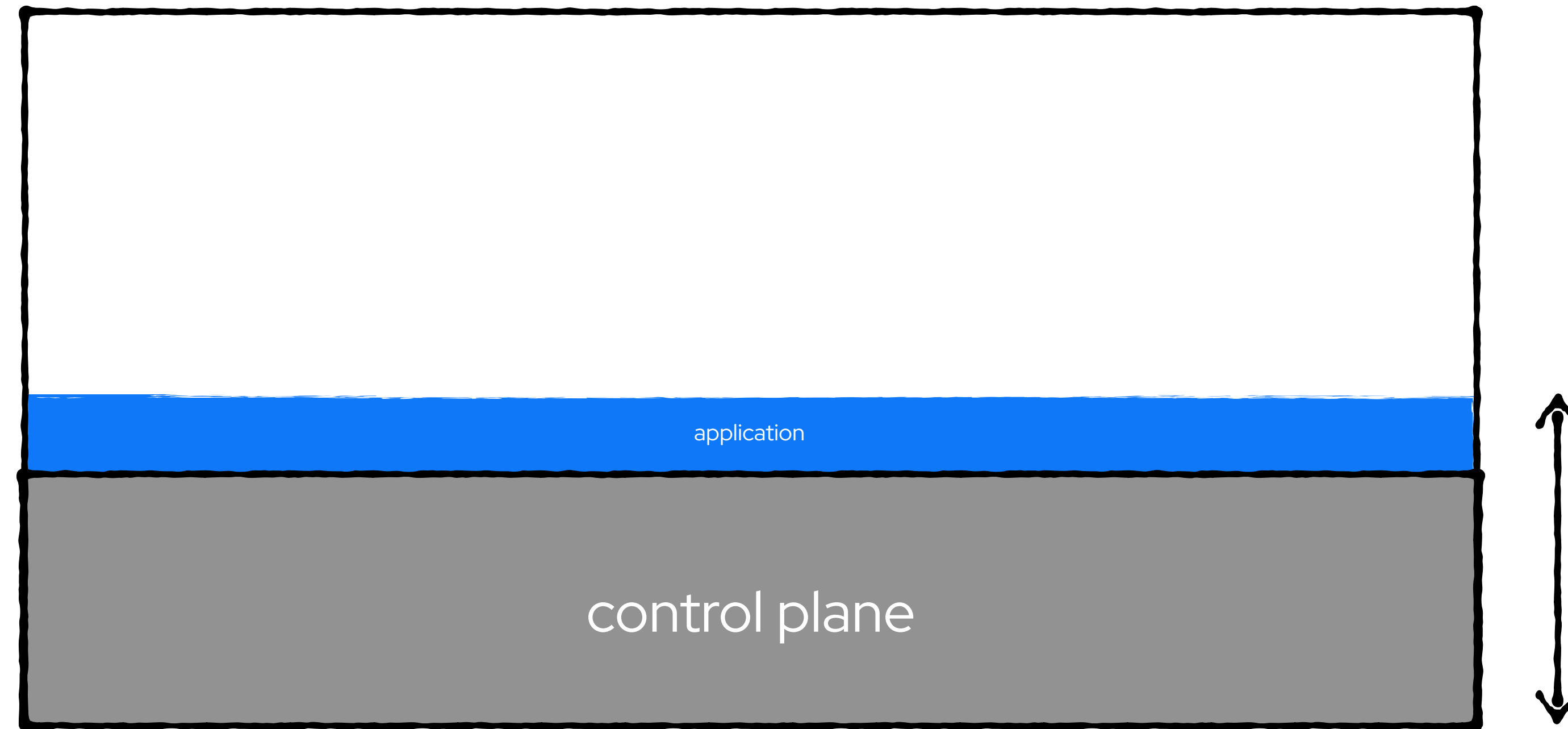
serverless systems may have high overheads



serverless systems may have high overheads



serverless systems may have high overheads



## Challenges and Opportunities in Sustainable Serverless Computing

*Prateek Sharma*  
*Indiana University Bloomington*  
*prateeks@iu.edu*

### Abstract

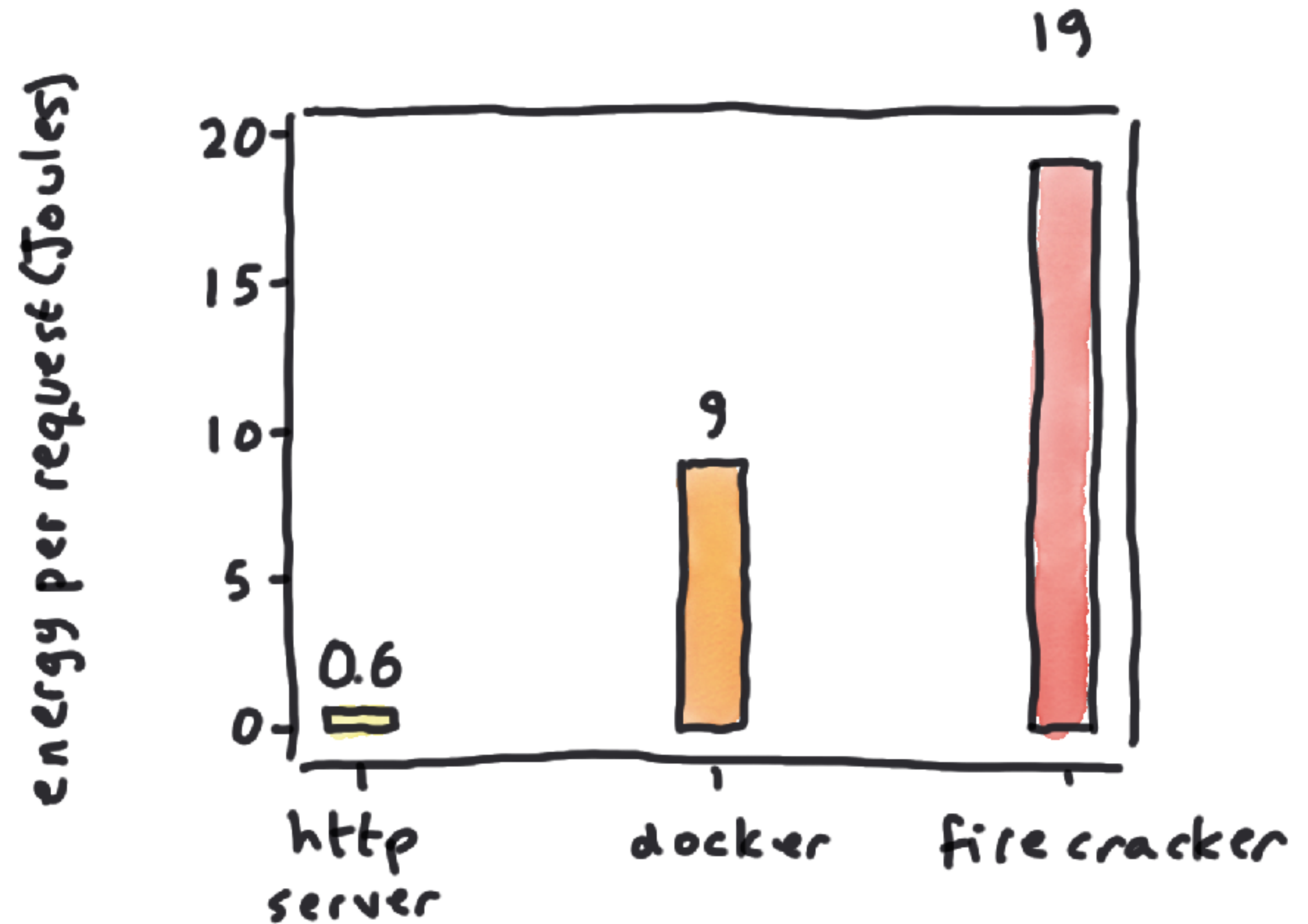
Serverless computing has rapidly emerged as a popular deployment model. However, its energy and carbon implications are unclear and require exploration. This paper takes a look at the fundamental distinguishing attributes of serverless functions, and shows how some of them make energy-efficiency challenging. The programming model and deployment requirements of serverless functions makes them terribly energy inefficient—consuming more than  $15\times$  energy compared to conventional web services. On the bright side, FaaS is still actively expanding, and there is also an opportunity for rethinking FaaS resource management and deployment models, and make carbon efficiency a primary consideration. We present a

and scientific computing, now use Functions as a Service (FaaS) offerings of cloud platforms such as Amazon Lambda, Azure and Google Functions, etc.

Given the sharp rise in its popularity, *what are the energy and carbon implications of FaaS?* While serverless computing has many benefits for *applications*, its programming model has imposed many resource management and optimization challenges for *FaaS providers* [28]. In the first part of this paper, we explore some of the key energy challenges that are a fundamental derivative of the FaaS programming and deployment models.

Our preliminary empirical investigation suggests that FaaS applications can be up to  $15\times$  more energy hungry than conventional web services. This energy and carbon (in)efficiency is unfortunately a fundamental attribute of

virtualisation overheads  
mean each function request  
can use 30x more energy  
than a plain http server





are all parts of the system elastic?

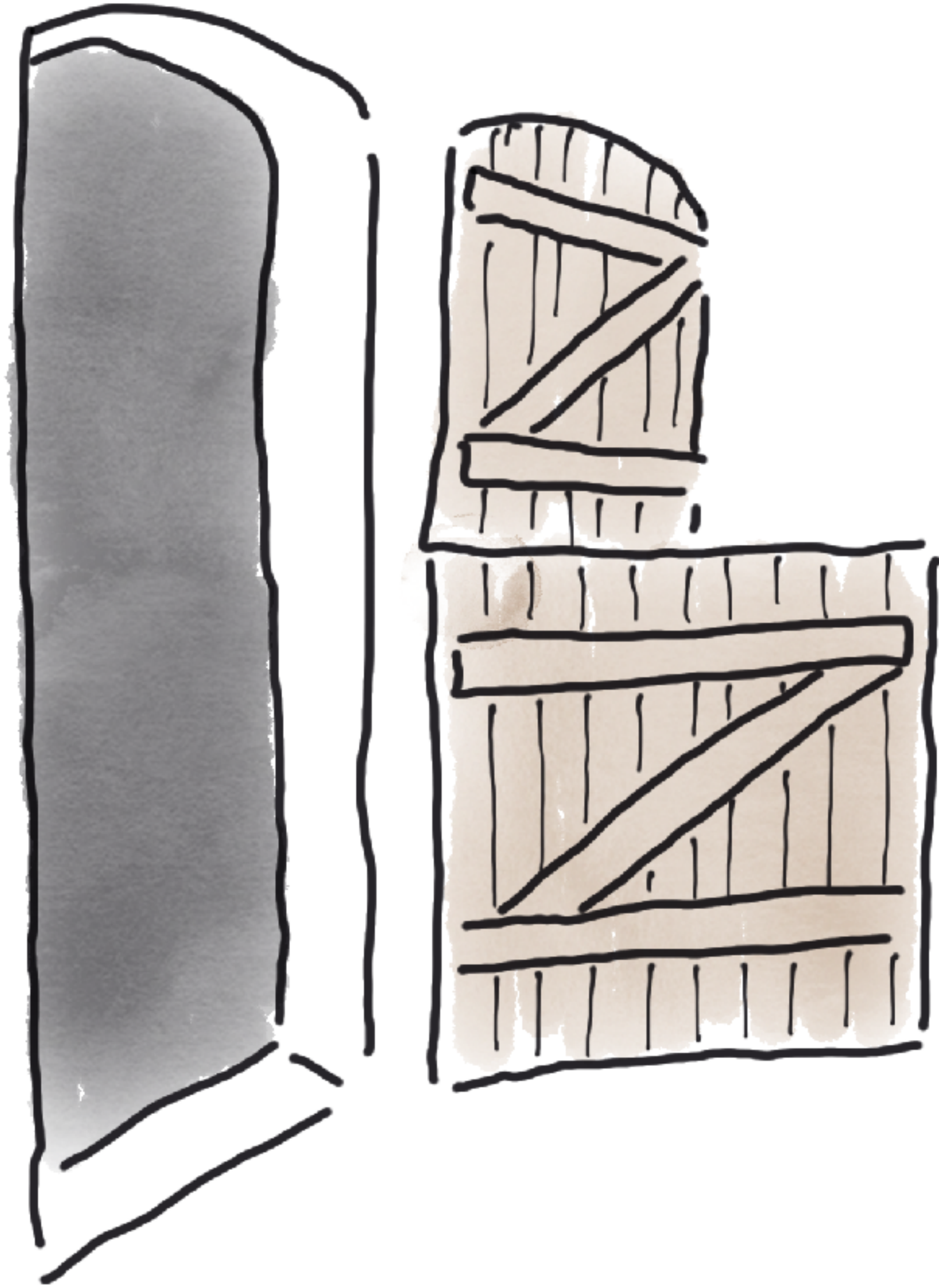
things that definitely don't help

things that don't help

prevention

things that don't help

prevention (?!)



surely shutting the barn door **before**  
the horse has left is a good idea?

prevention == heavy governance

remember the ikea effect?

remember the ikea effect?

people will not surrender

servers that were hard to get



zombies are not just servers

data

traffic

zombie packets

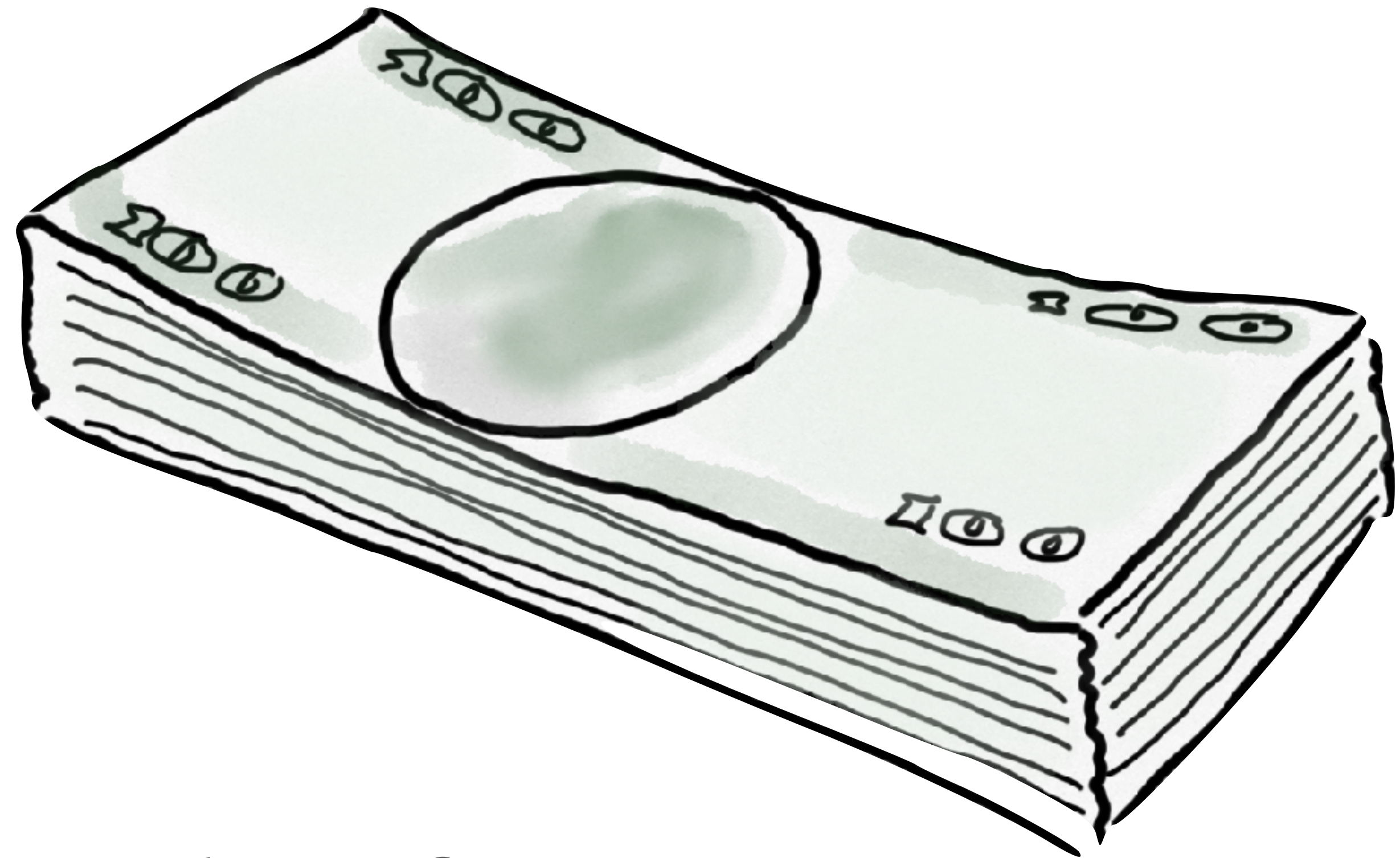
internet background noise

internet background noise

5.5 gigabits/s

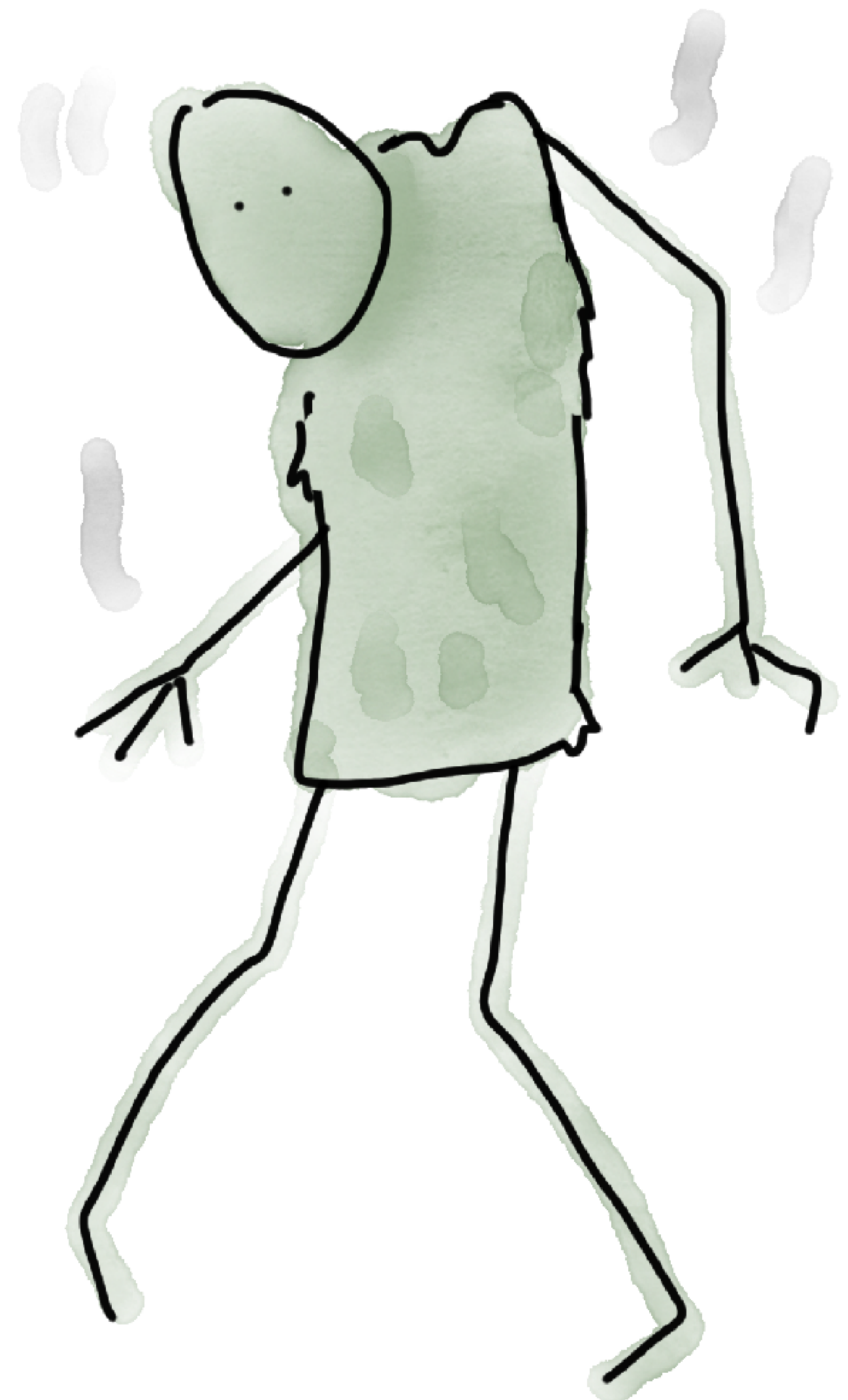
unsolved problem == opportunity

# the double-win

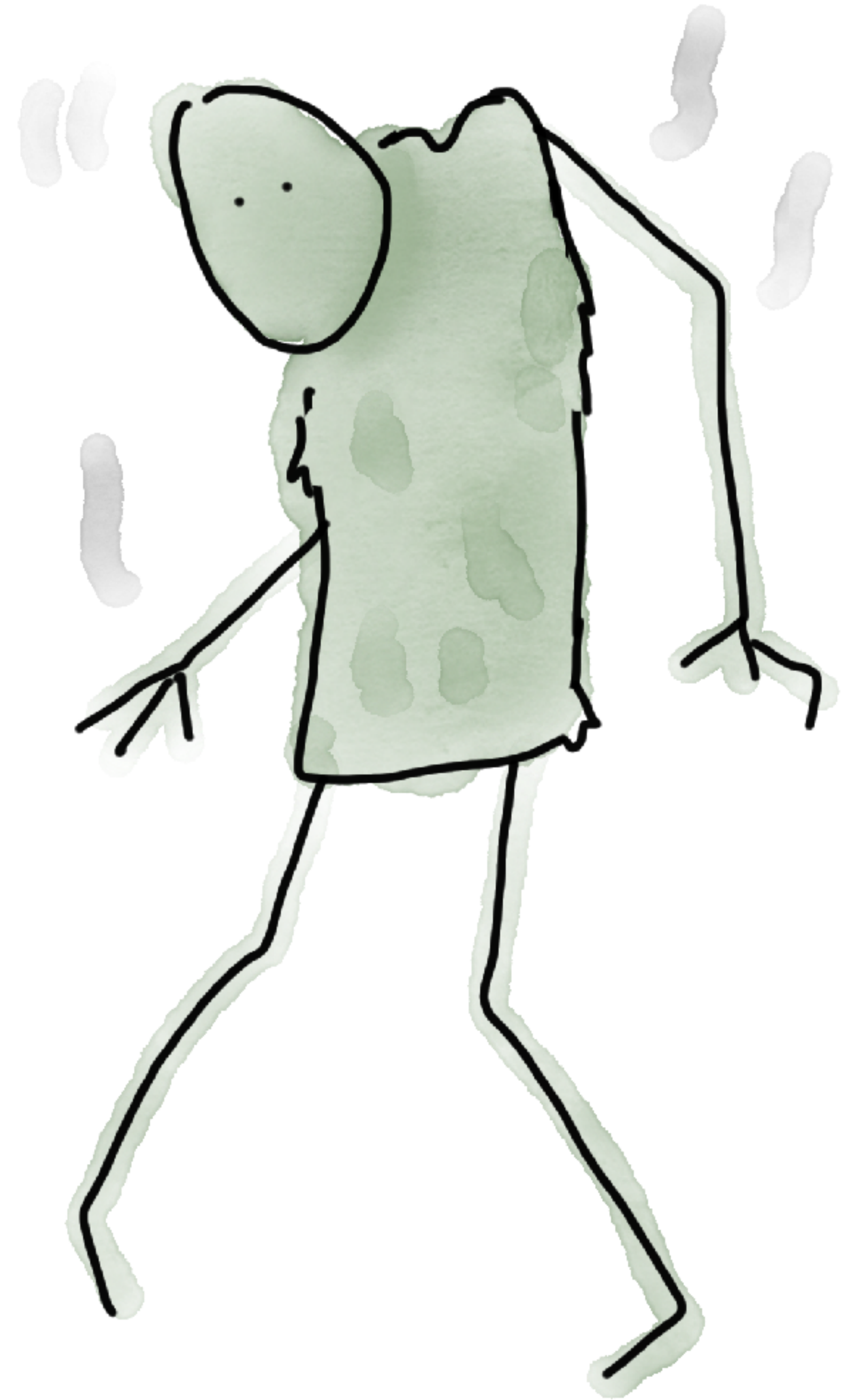


turning things off saves a lot of money





users ...



users ...

up utilisation

aim for elasticity

limit kubesprawl

de-zombify

know what you're using

turn it off





tool creators, support



# tool creators, support



better utilisation

elasticity

multi-tenancy

de-zombification

visibility

disposability

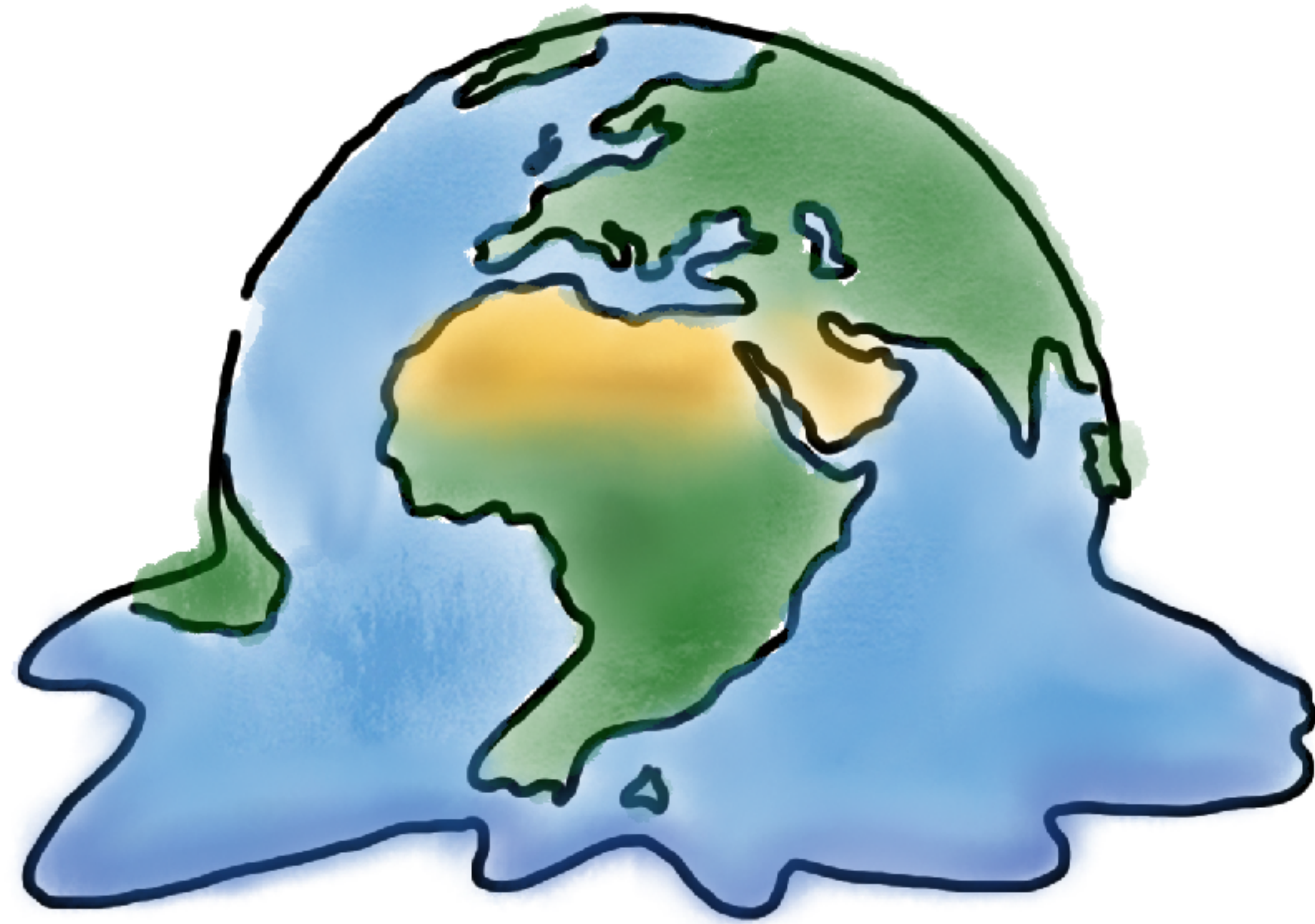
GreenOps

FinOps

AI Ops

GitOps

LightSwitchOps



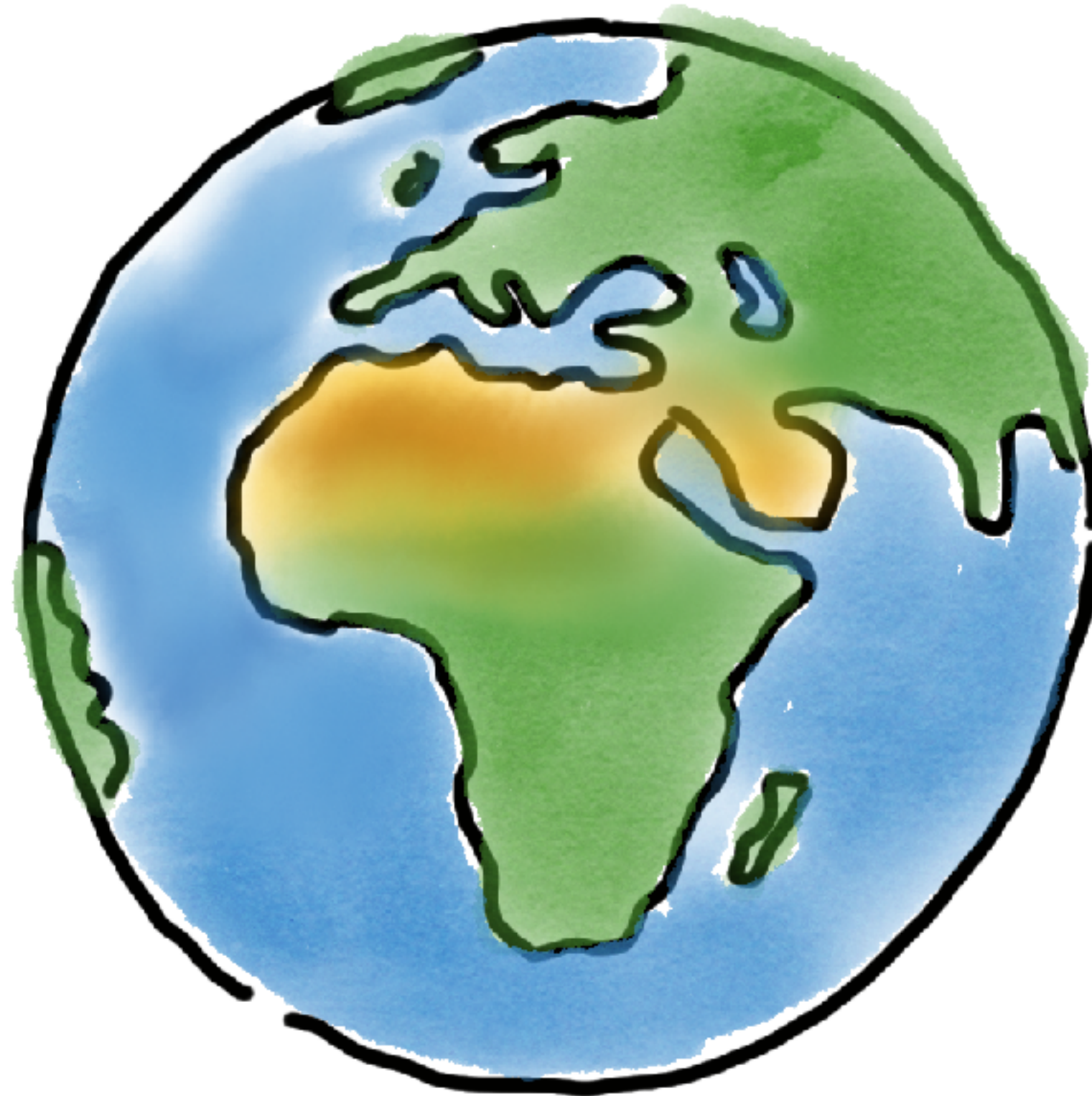
GreenOps

FinOps

AI Ops

GitOps

LightSwitchOps





thank you

@holly\_cummins@hachyderm.io



slides

