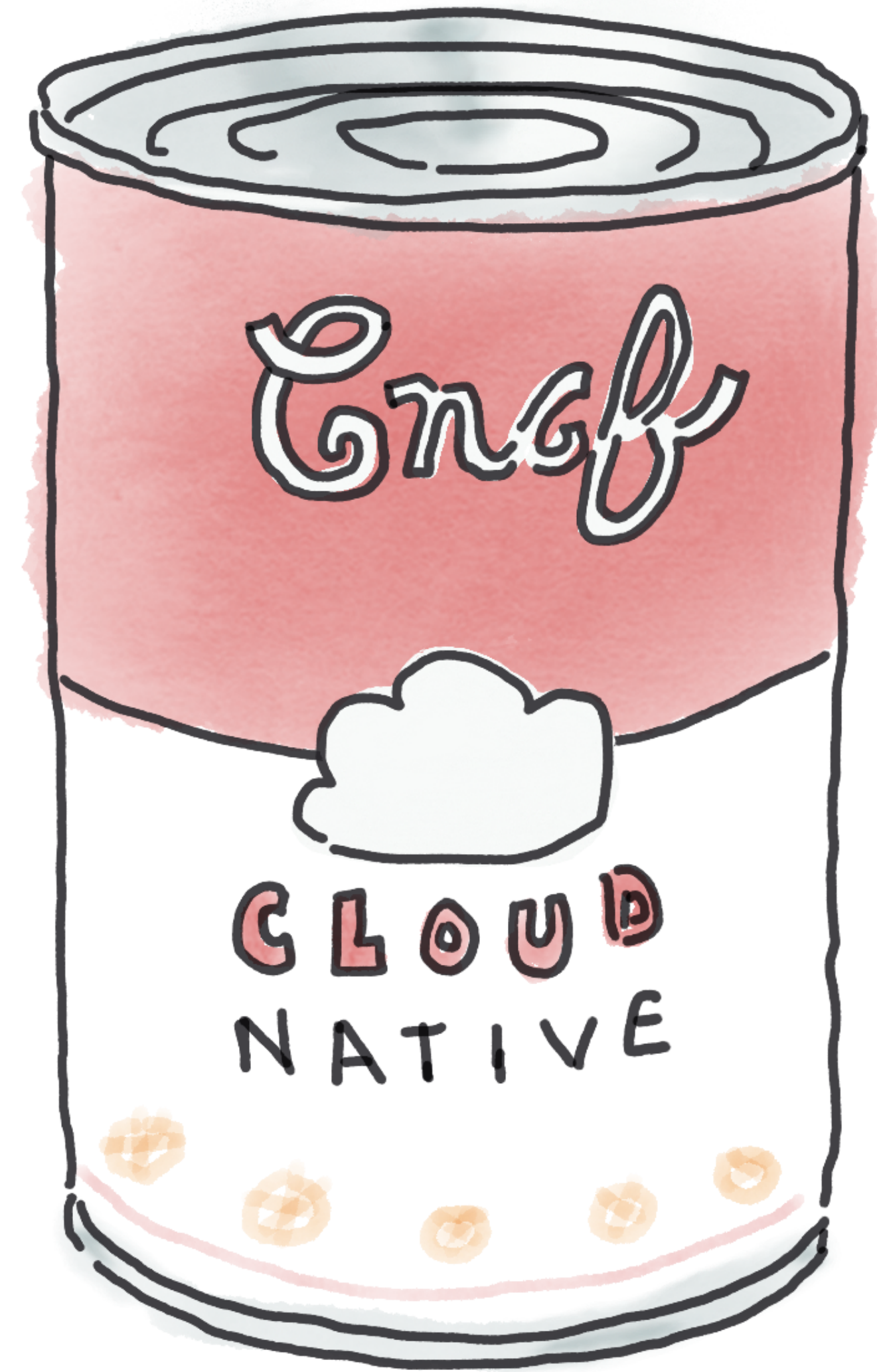


# why you can't buy cloud native

Holly Cummins  
**IBM**

@holly\_cummins







**Sam Newman** ✓  
@samnewman



Interesting side effect of the Cloud Native Foundation is now that I'm commonly speaking to people who believe it can't be cloud native without Kubernetes 🙄

10:38 PM · Nov 19, 2020 · Tweetbot for iOS

**15** Retweets   **4** Quote Tweets   **130** Likes

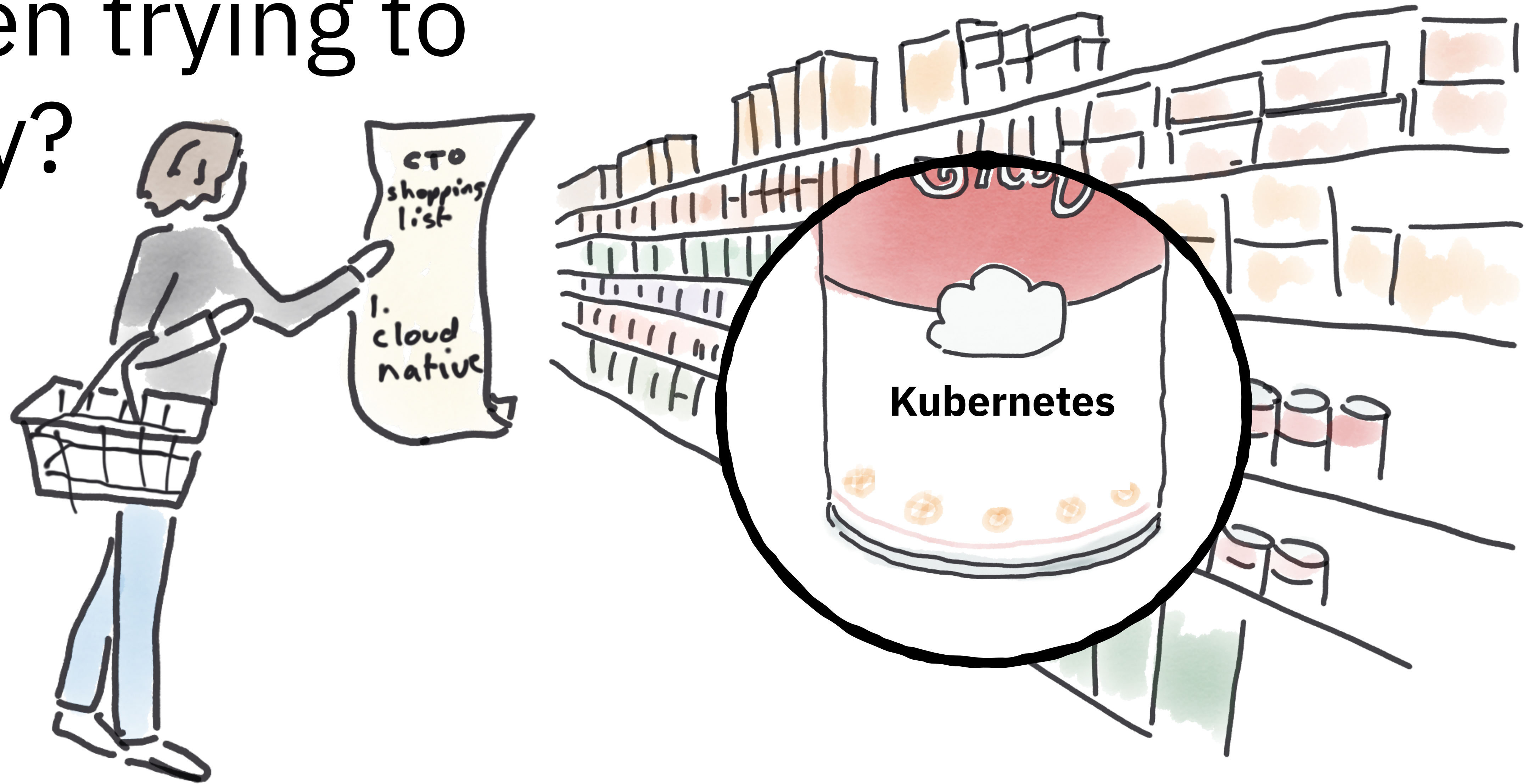
what are we  
even trying to  
buy?



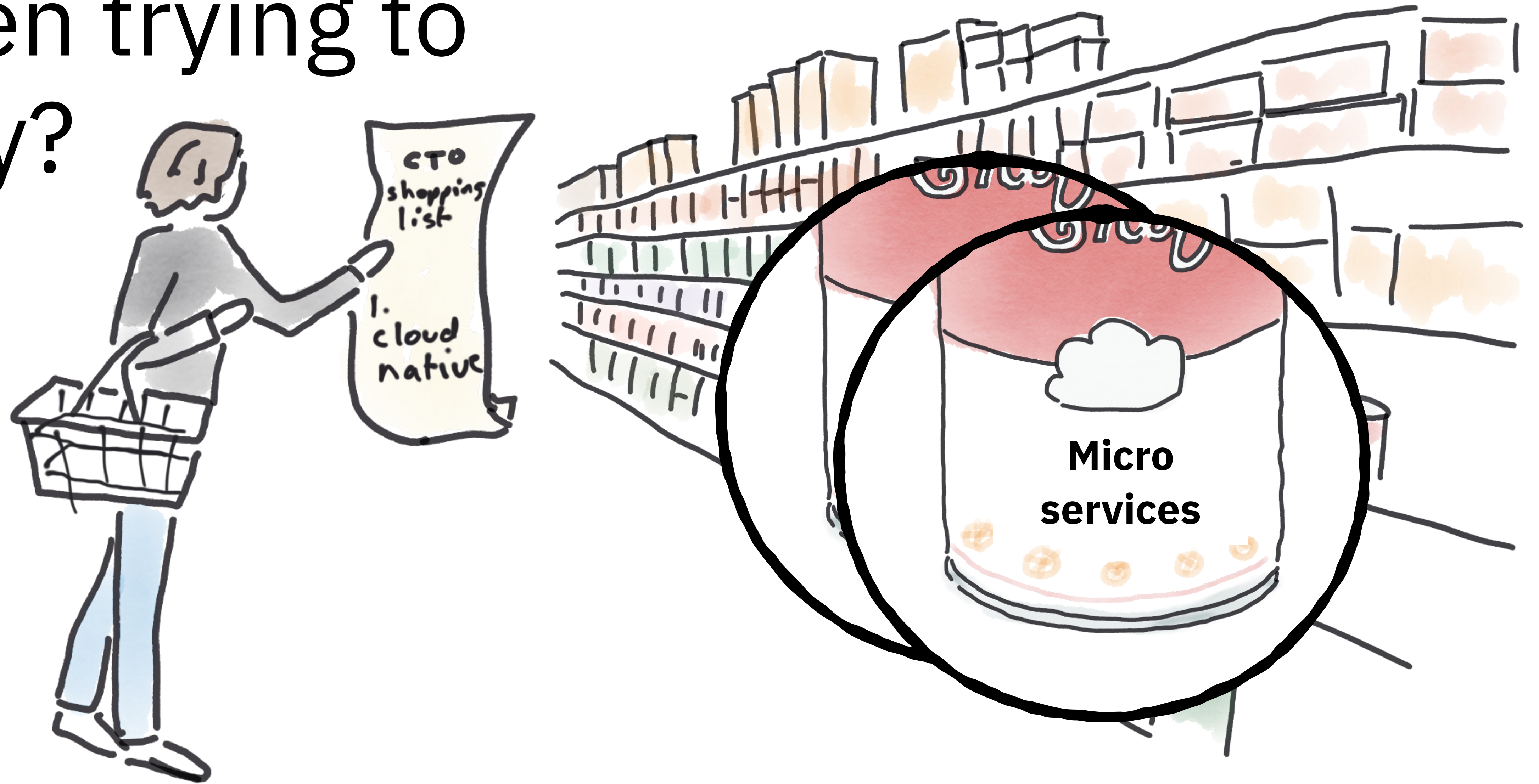
what are we  
even trying to  
buy?



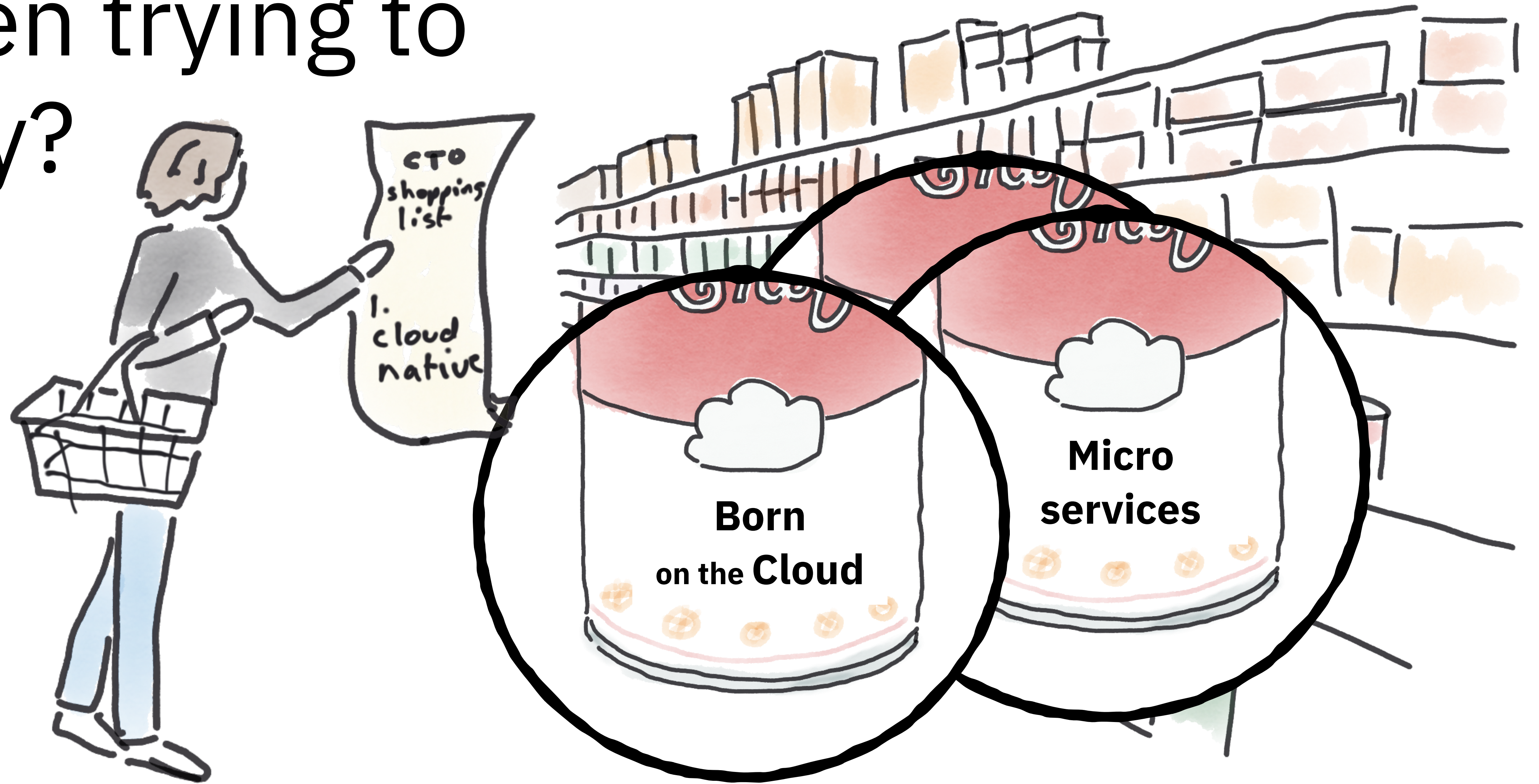
what are we  
even trying to  
buy?



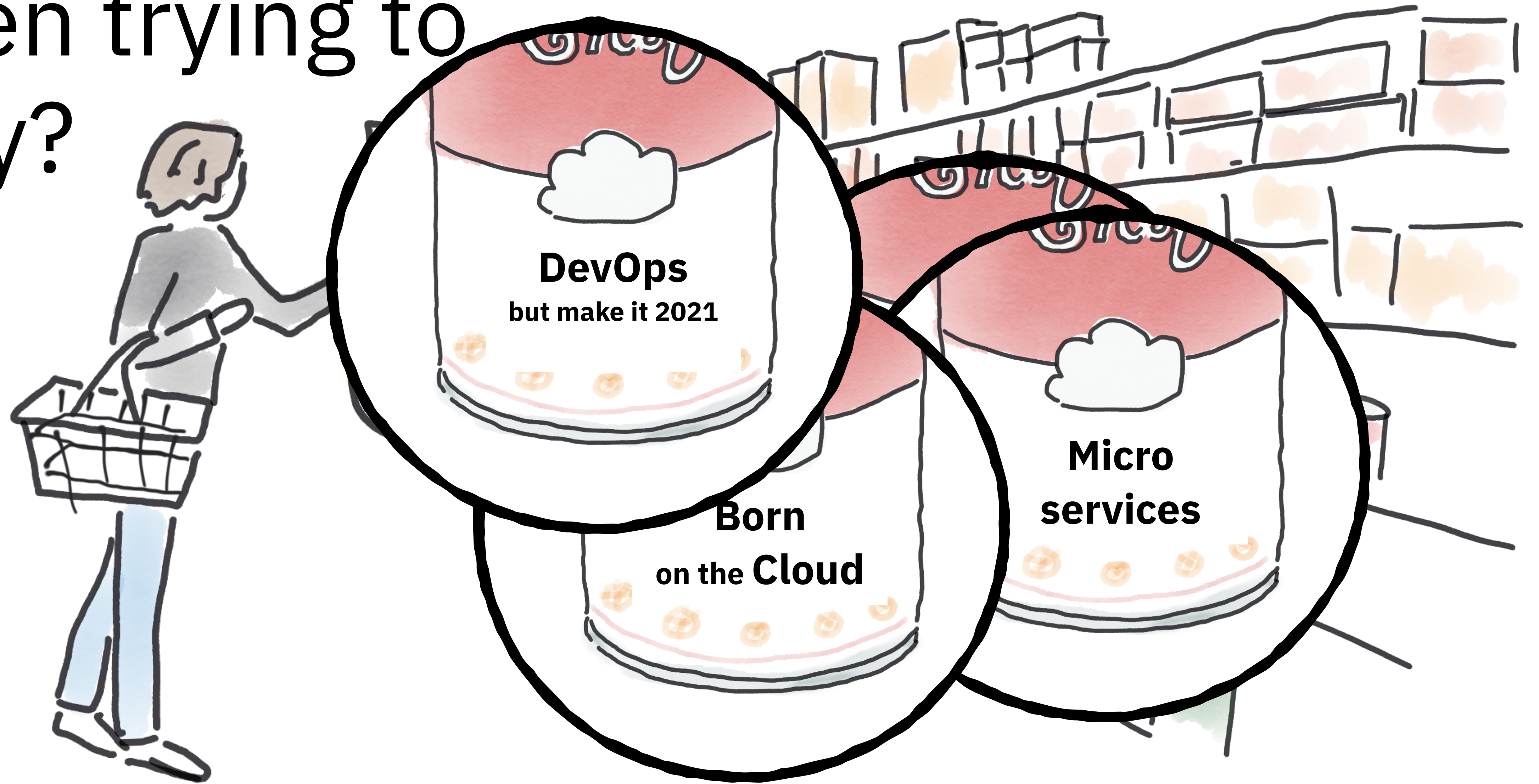
what are we  
even trying to  
buy?



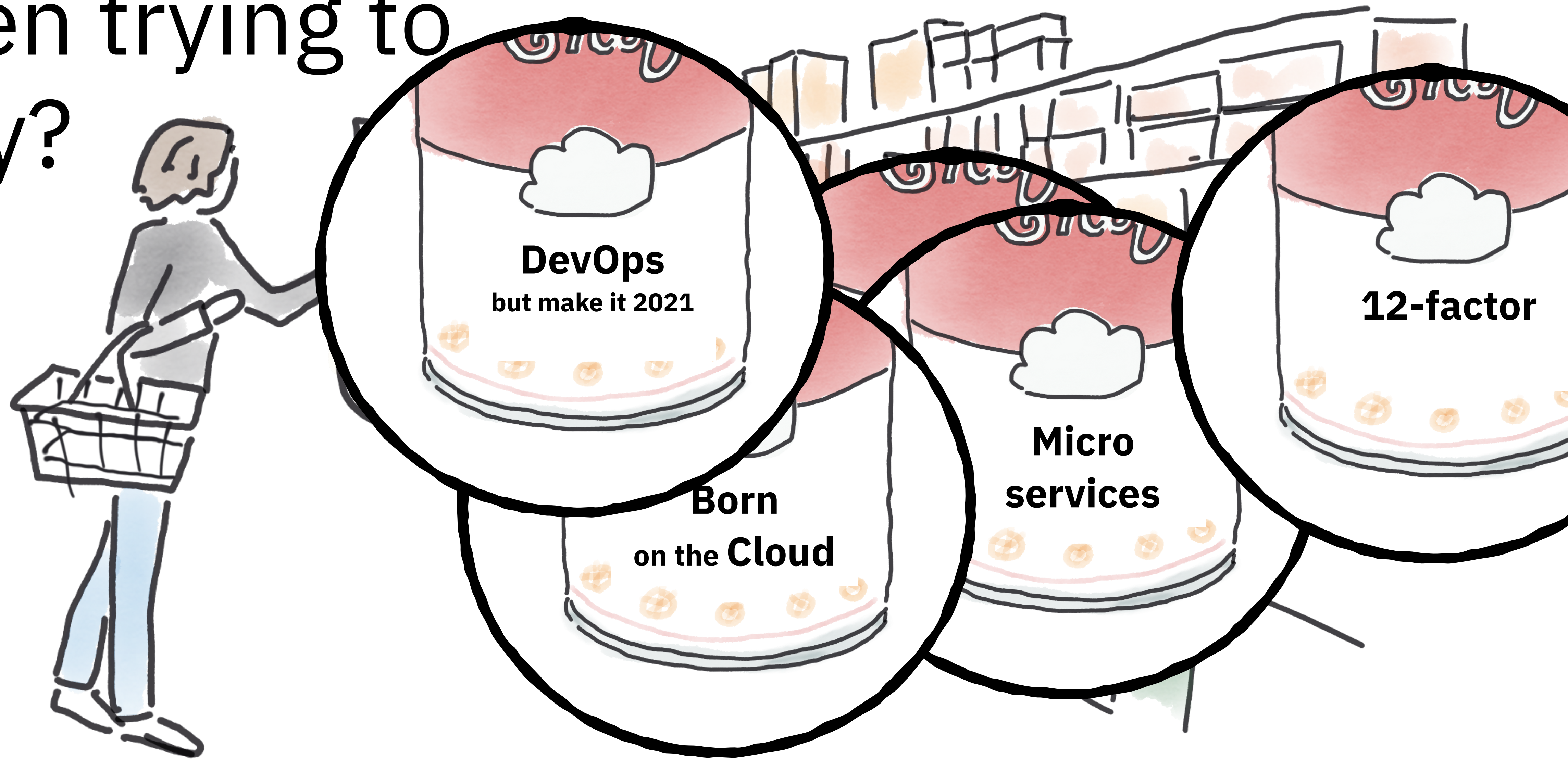
what are we  
even trying to  
buy?



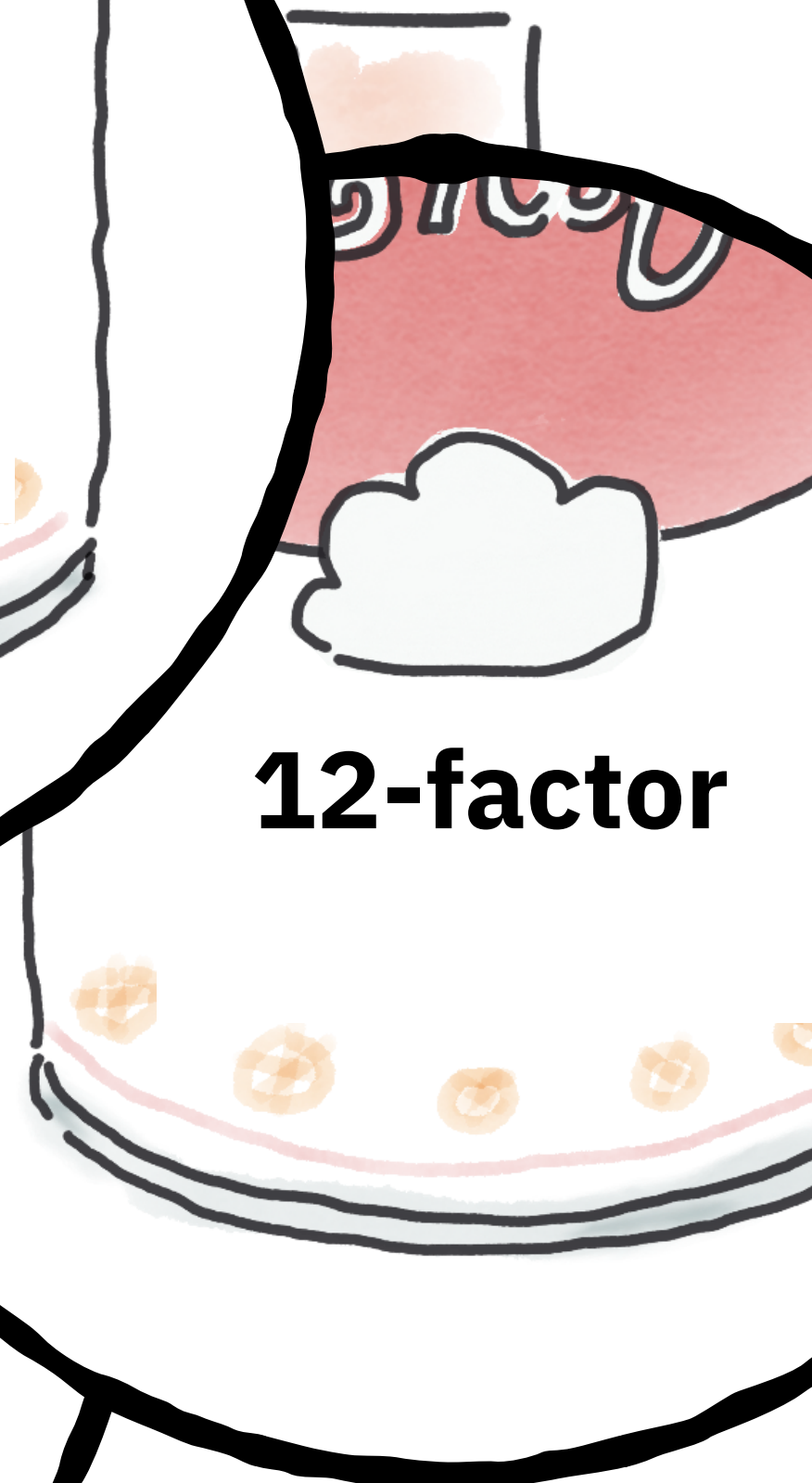
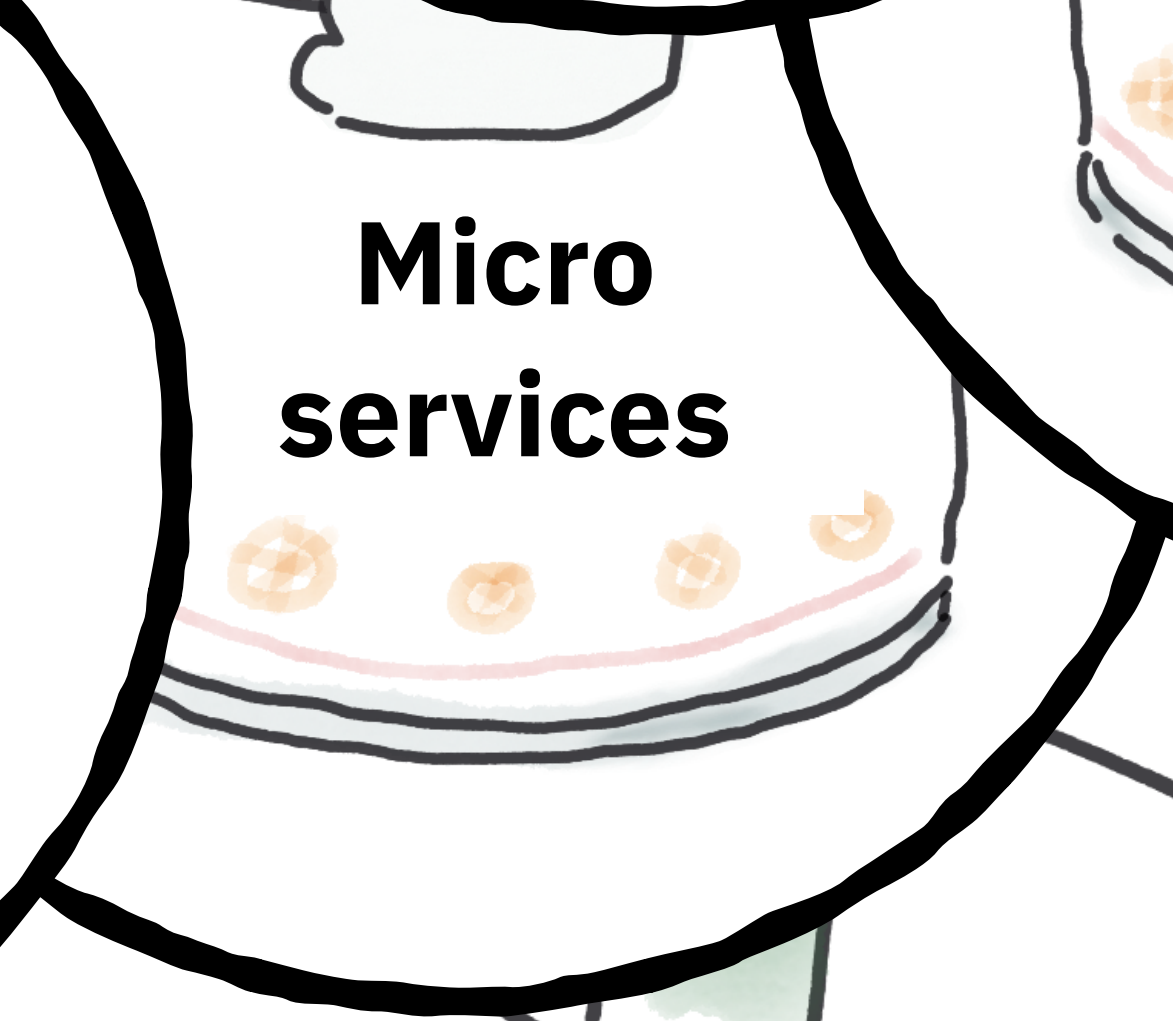
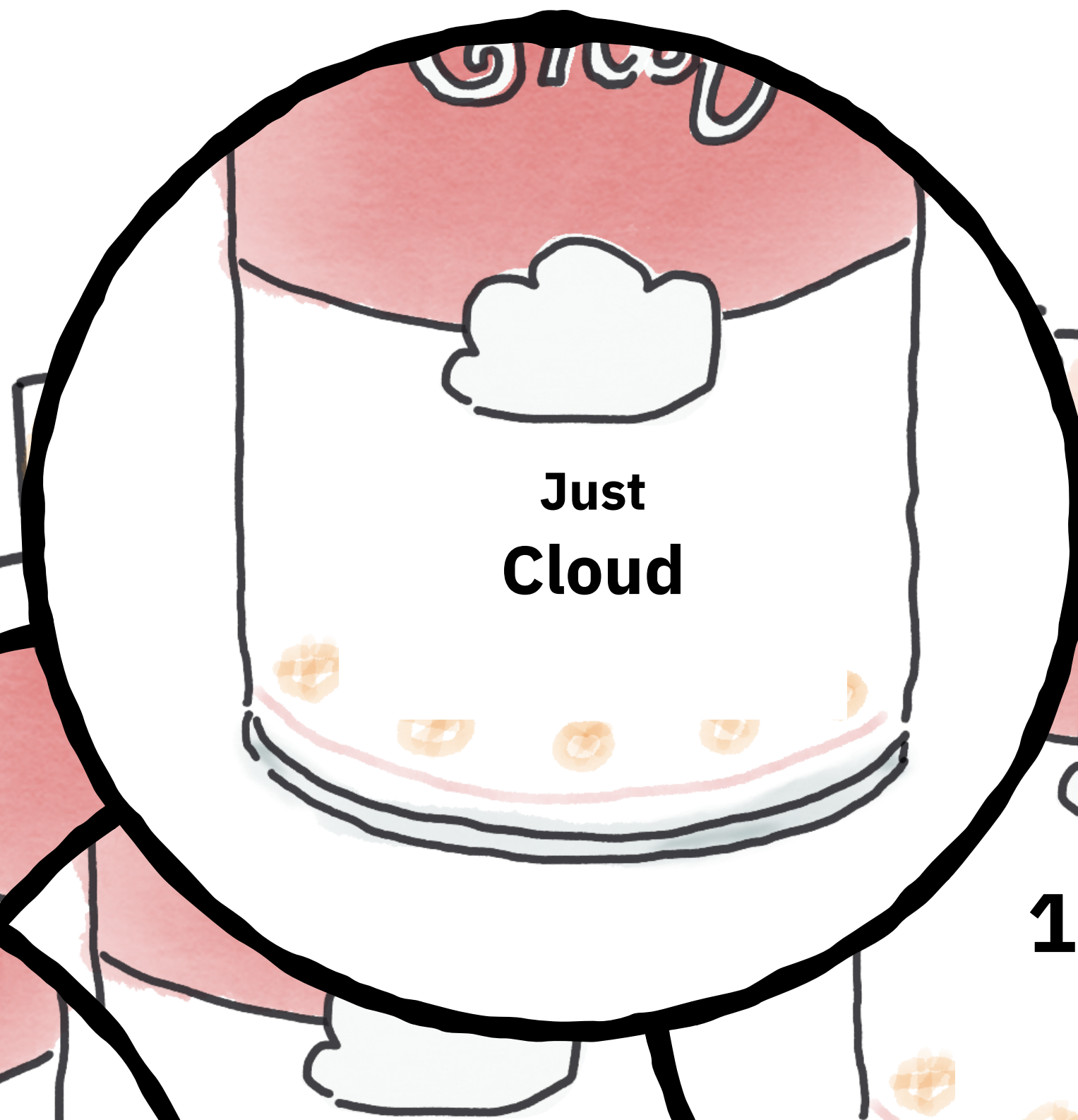
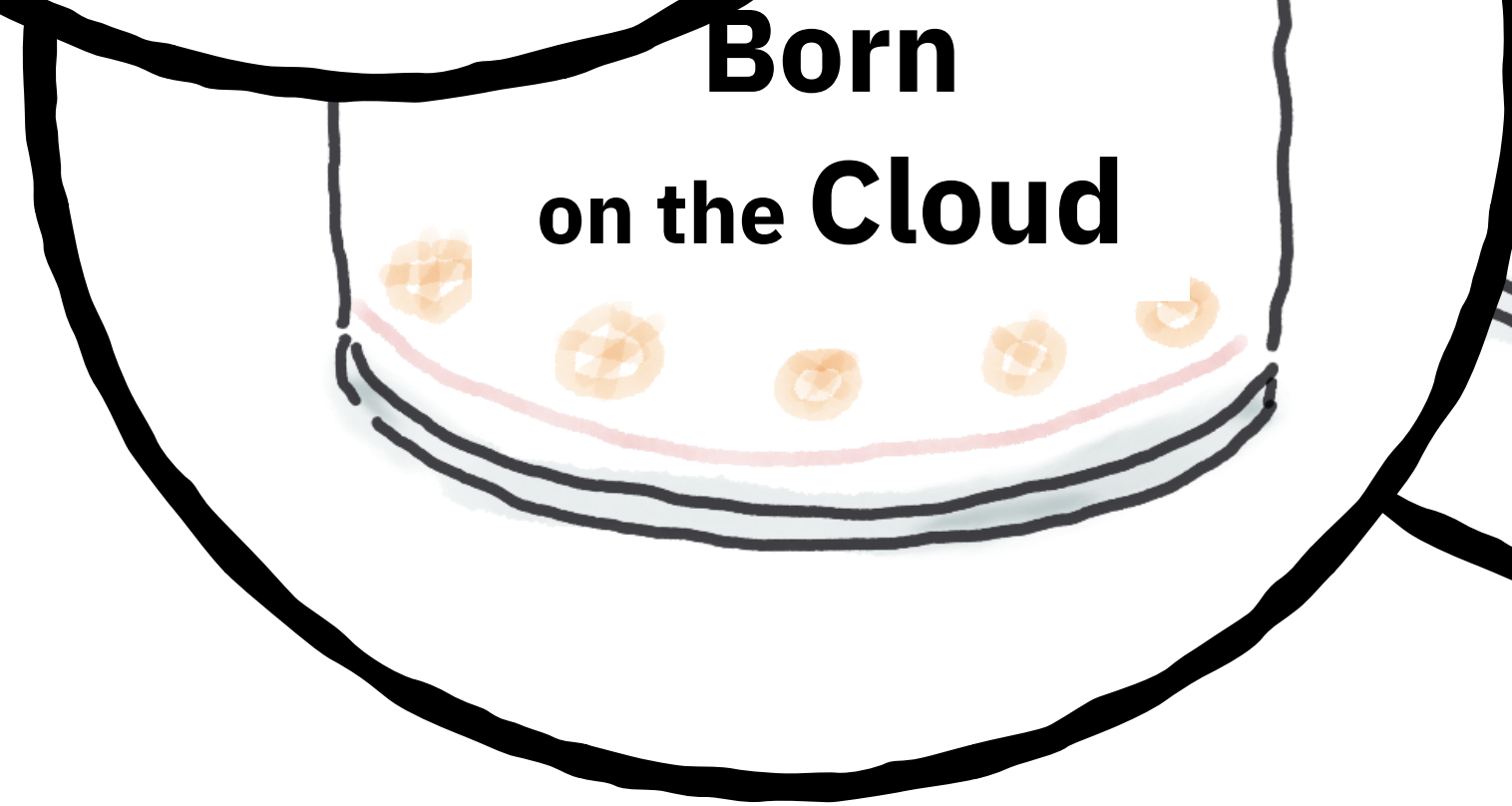
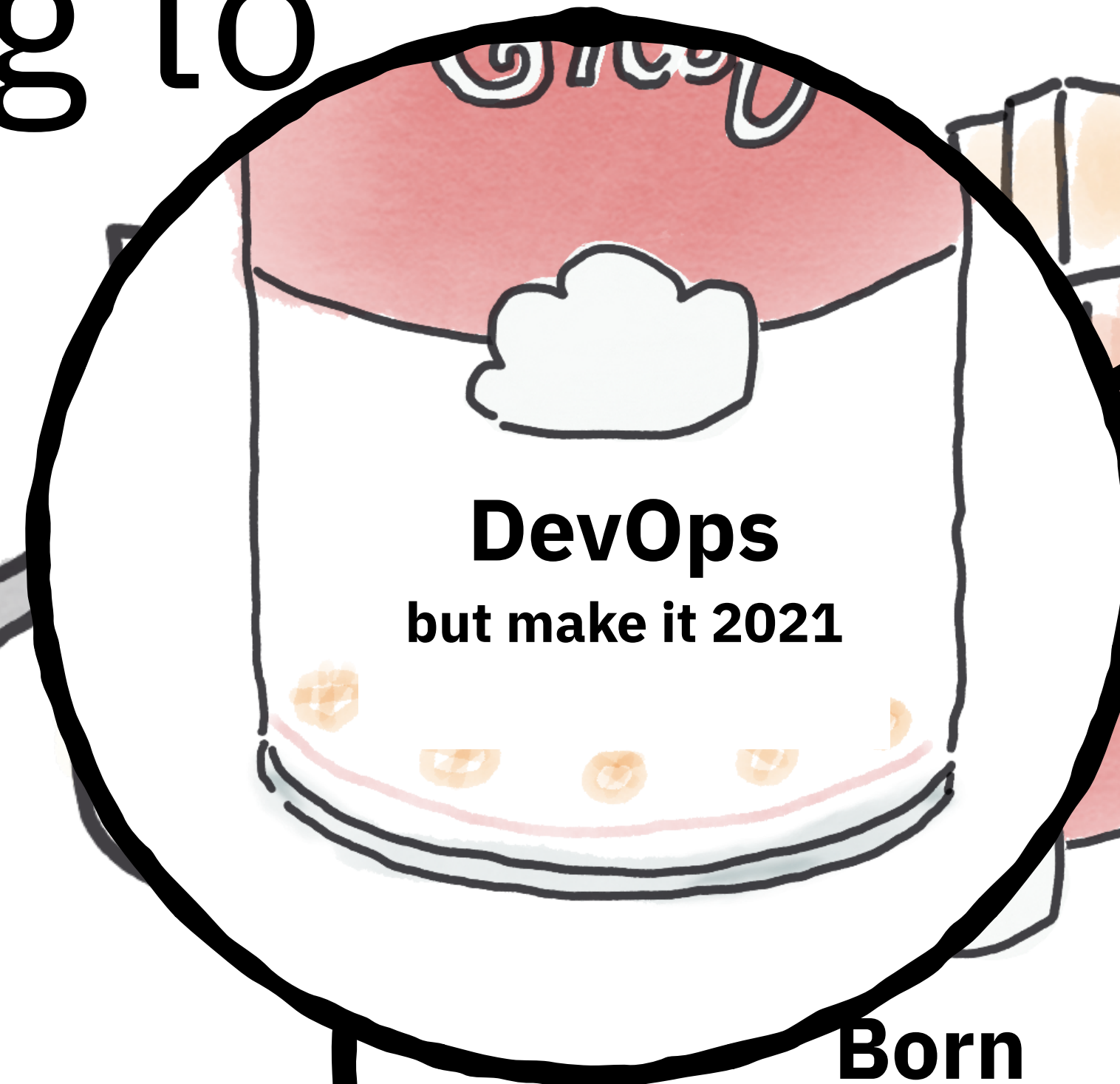
what are we  
even trying to  
buy?



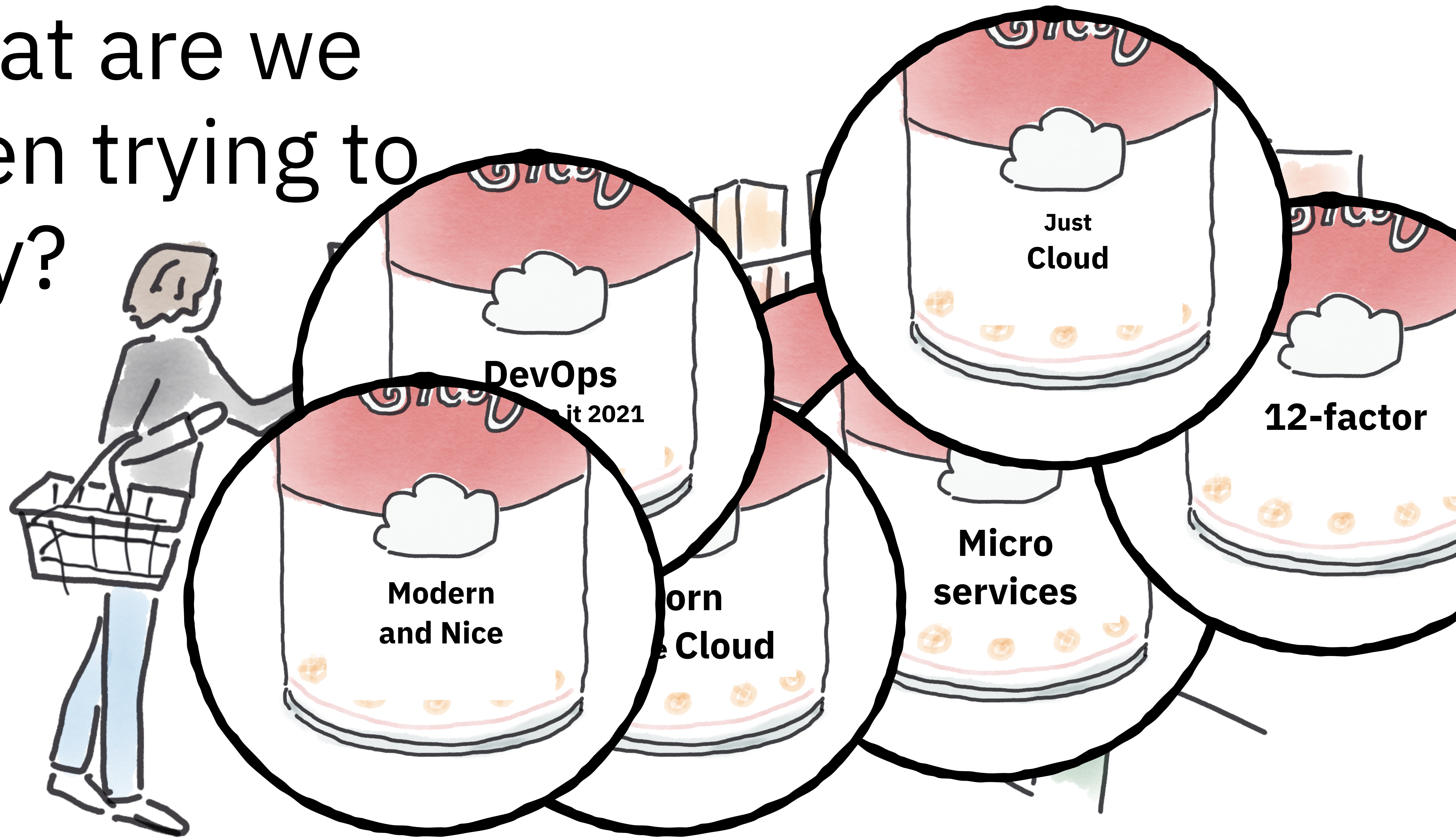
what are we  
even trying to  
buy?



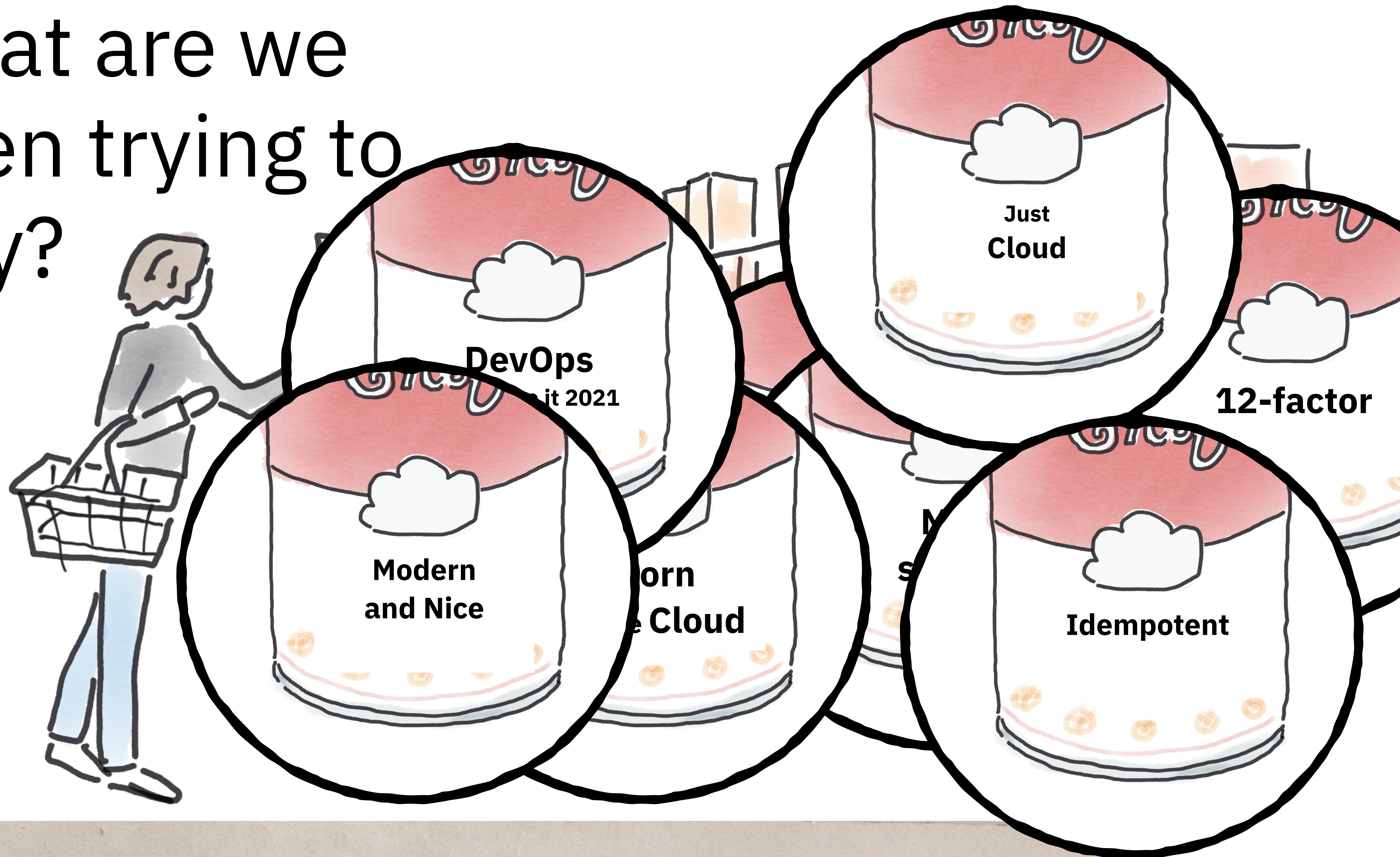
what are we  
even trying to  
buy?



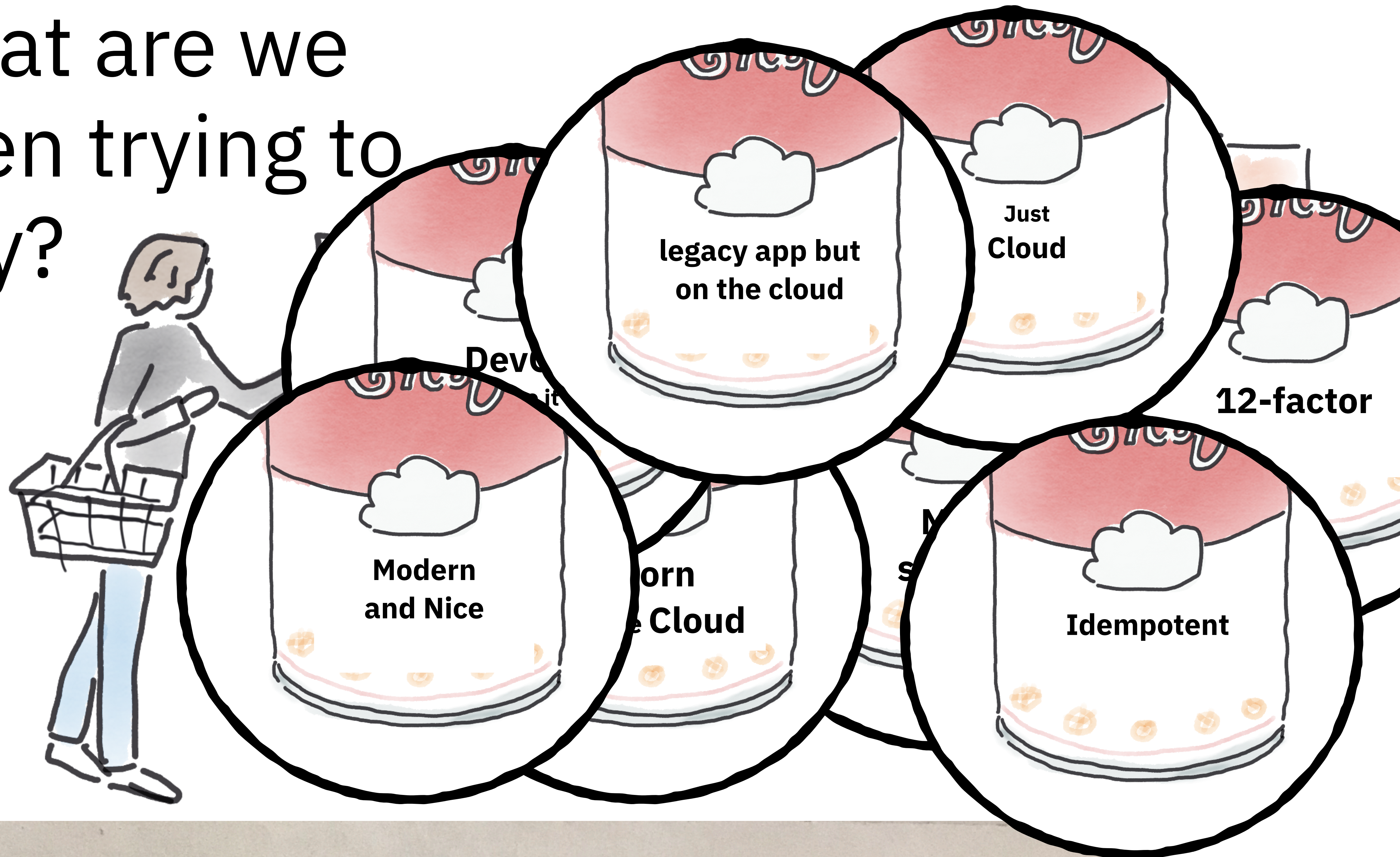
what are we  
even trying to  
buy?



what are we  
even trying to  
buy?



what are we  
even trying to  
buy?



what do you  
all think?



[#684983](https://app.sli.do/event/kcvns9iw)

# THE ARCHITECTURE SHOP

microservices  
buy 1 get 8 free



have you greenfield  
monolithed lately?



[#684983](https://app.sli.do/event/kcvns9iw)

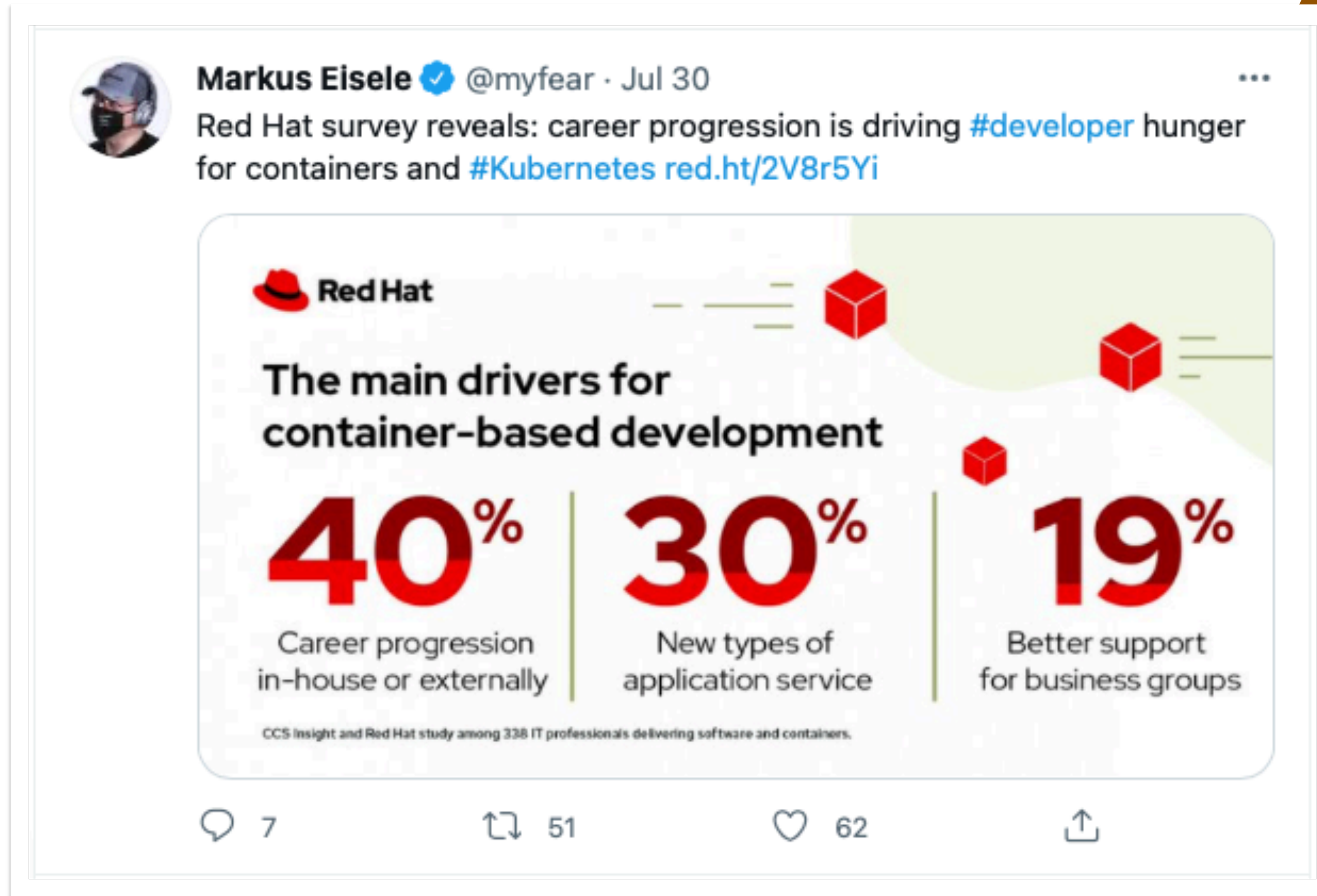
microservices  
will make me  
look **so** good



surely no one would ...

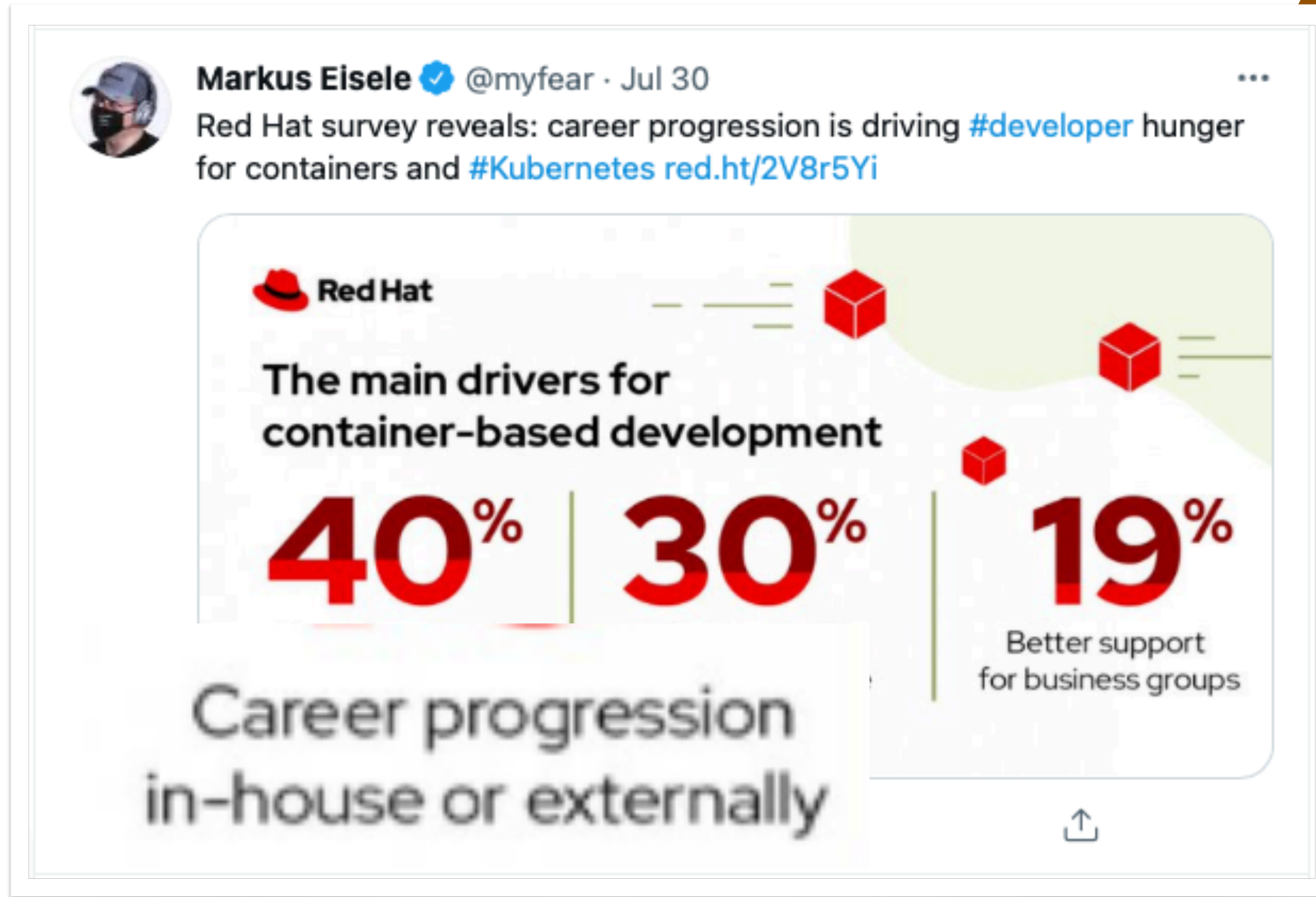


surely no one would ...



surely no one would ...

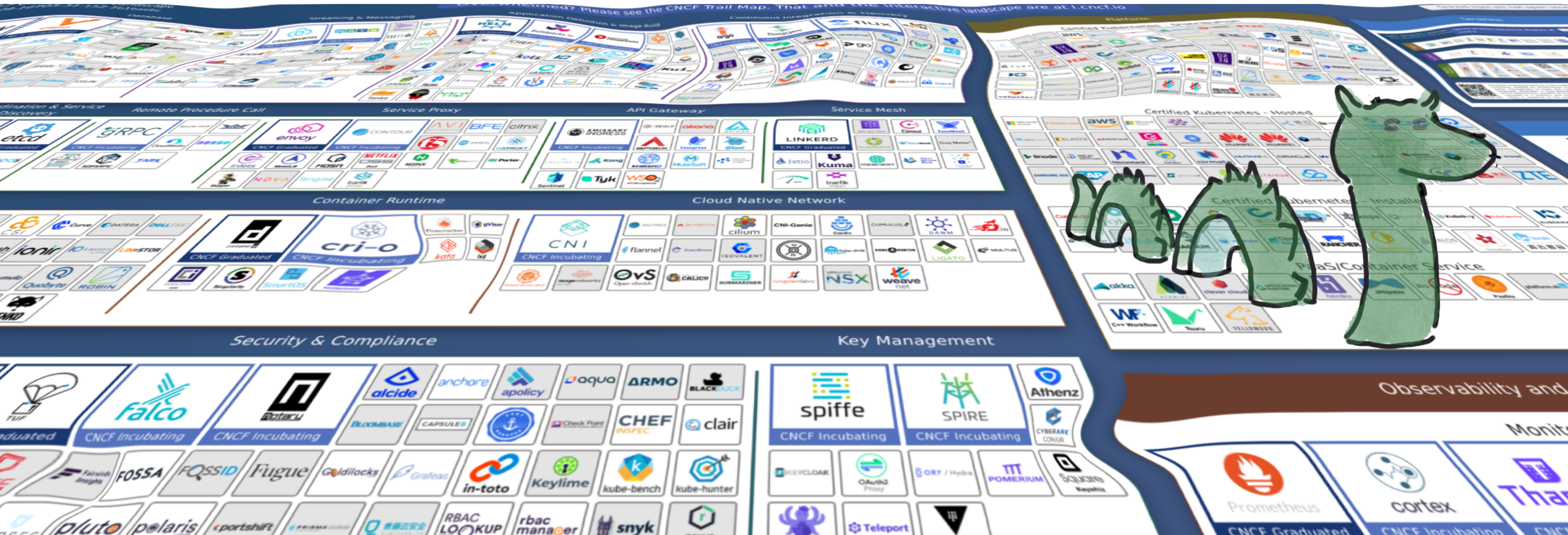
oh.



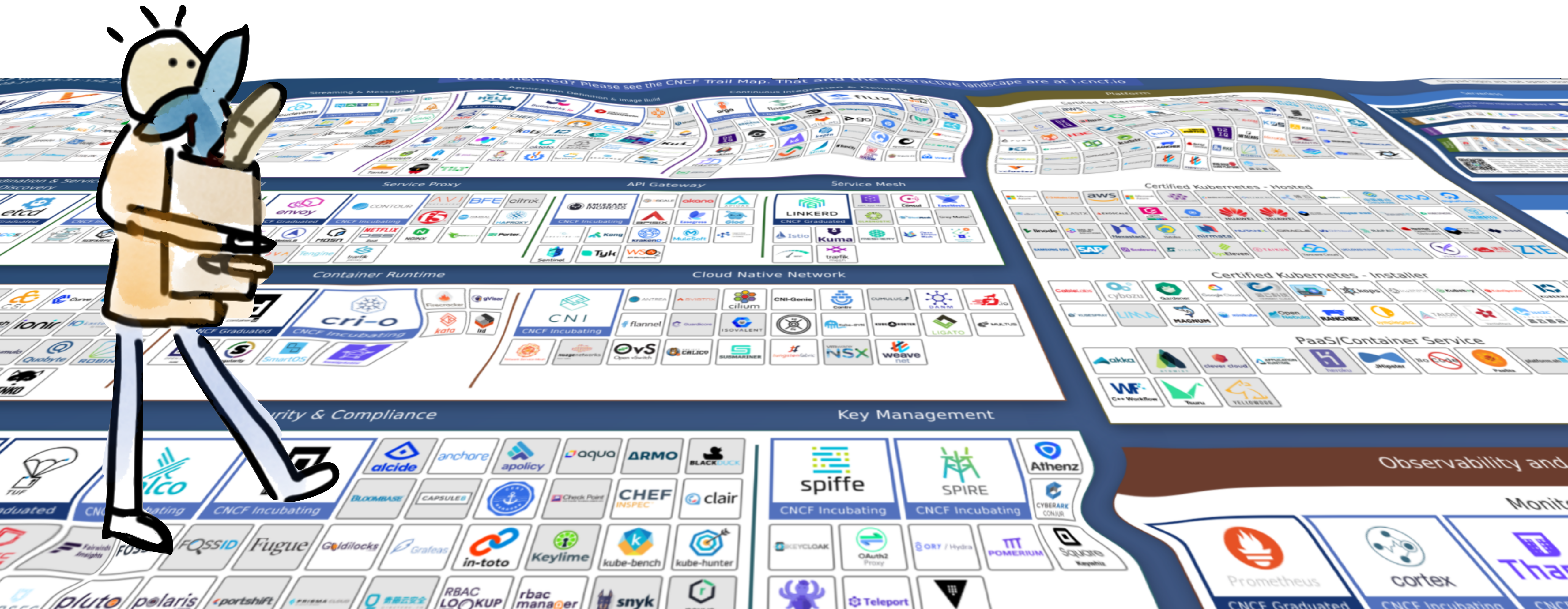
cv-driven  
development



there's a whole  
landscape to explore



it's not just a landscape  
it's a **catalog**



ooh, which  
ones shall I have?

not just a landscape  
it's a **catalog**



ooh, which  
ones shall I have?

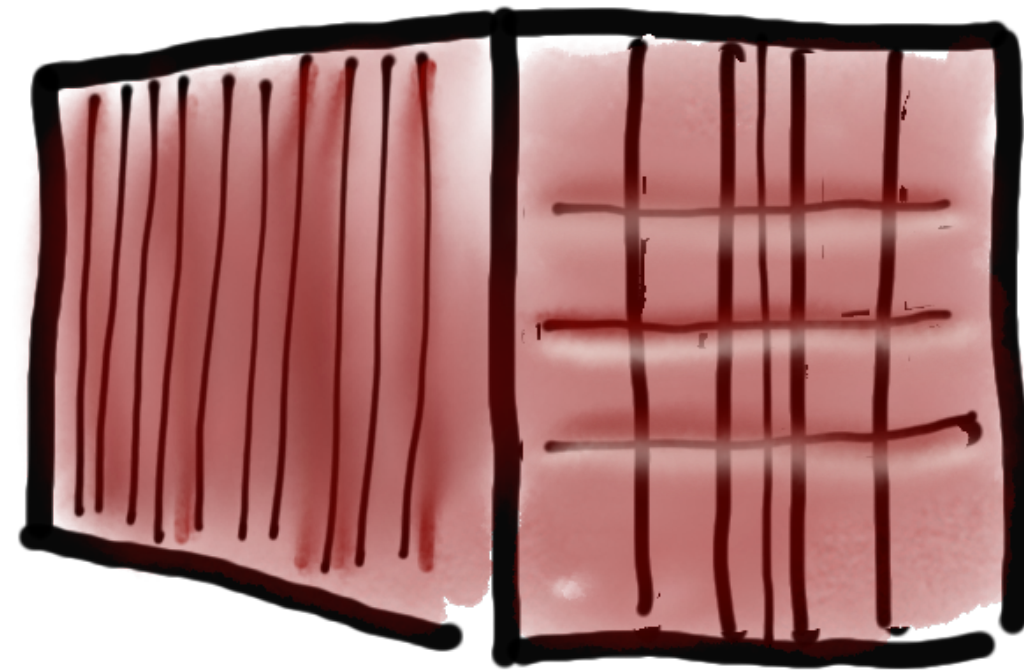
not just a landscape  
it's a **catalog**

they're **all**  
so shiny



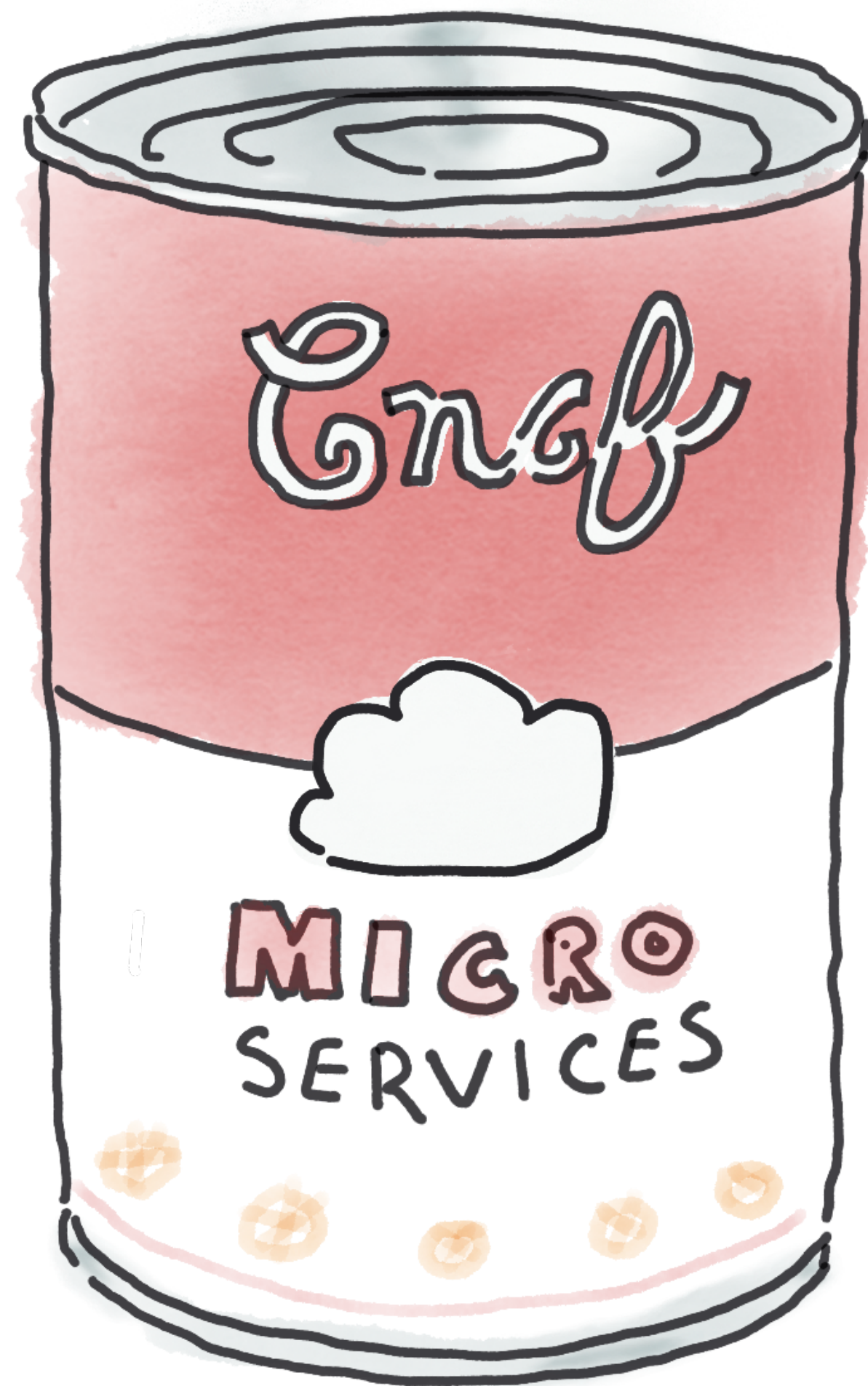
what problem are  
we trying to solve?

my problem is  
there aren't  
enough  
containers yet?

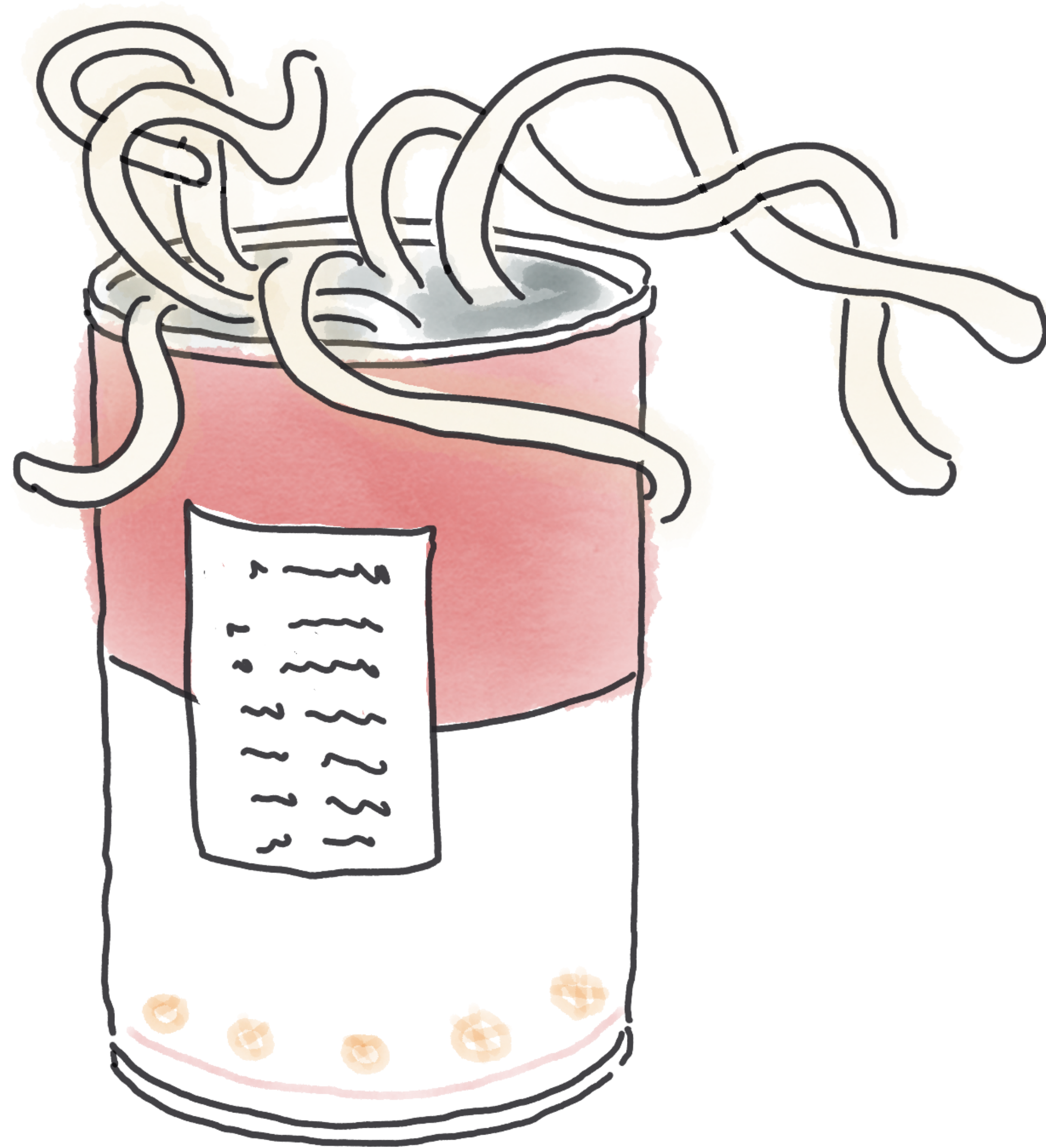


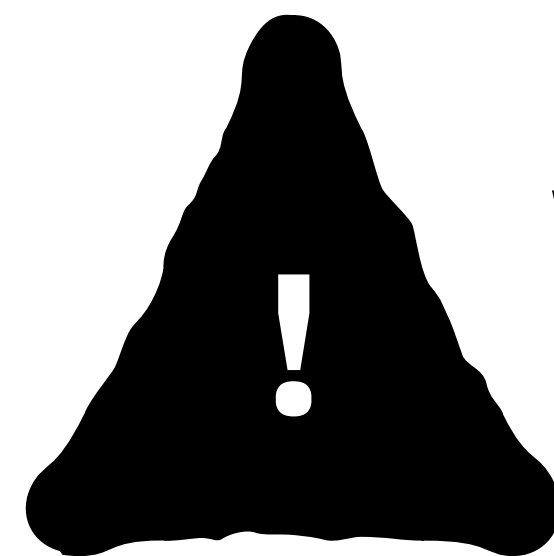
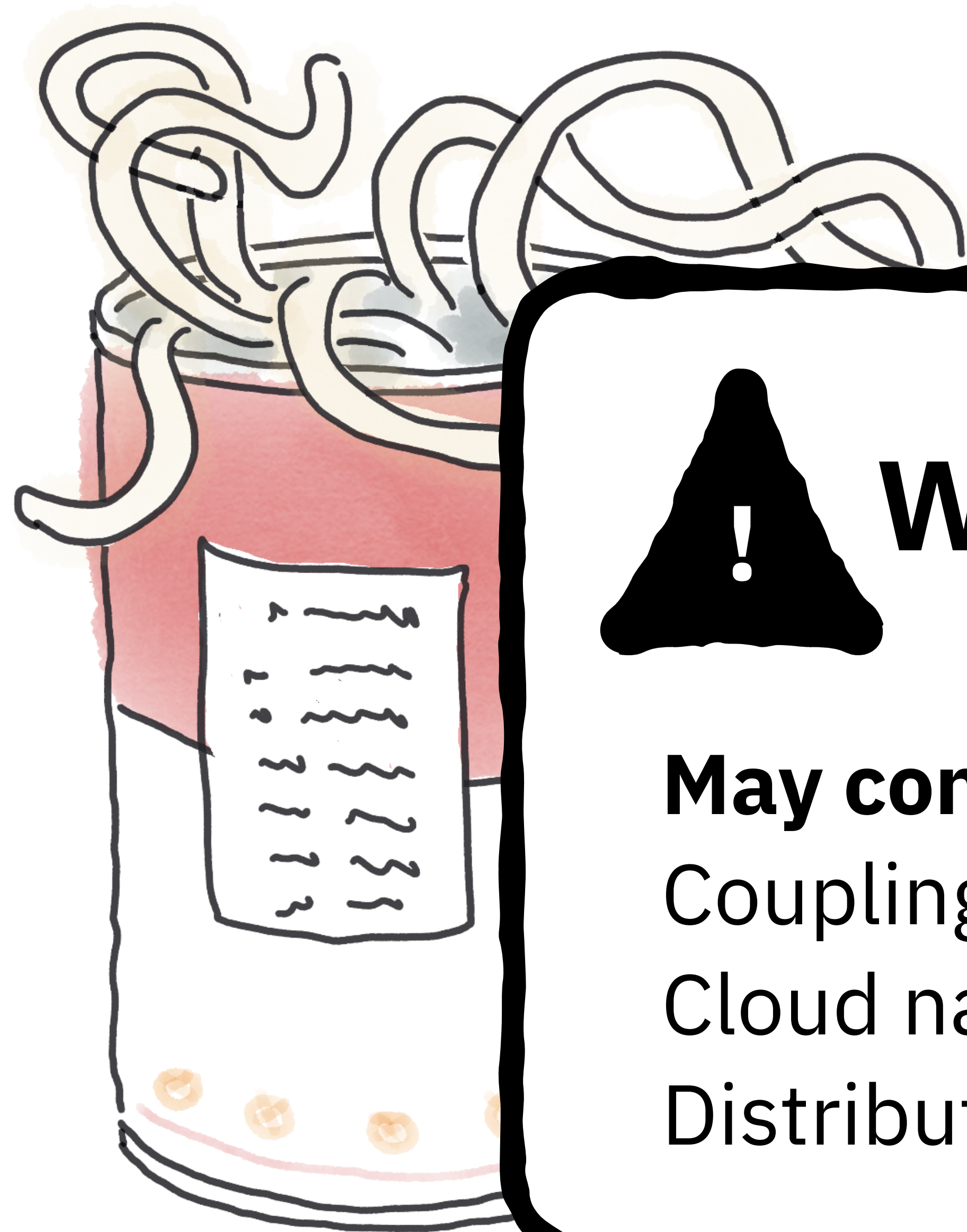


my problem is  
there aren't  
enough  
containers yet?









## Warning

**May contain**

Coupling

Cloud native spaghetti

Distributed monolith

♥ Molly Struve 🦄 and Camille Fournier liked



**Kelly Shortridge** ✓ @swagitda\_ · 14h



OH: "Do you have microservices or do you have a monolith spread over hundreds of git repos?"

💬 14

↻ 69

♥ 370



# distributed monolith

# distributed monolith

worst of both worlds:

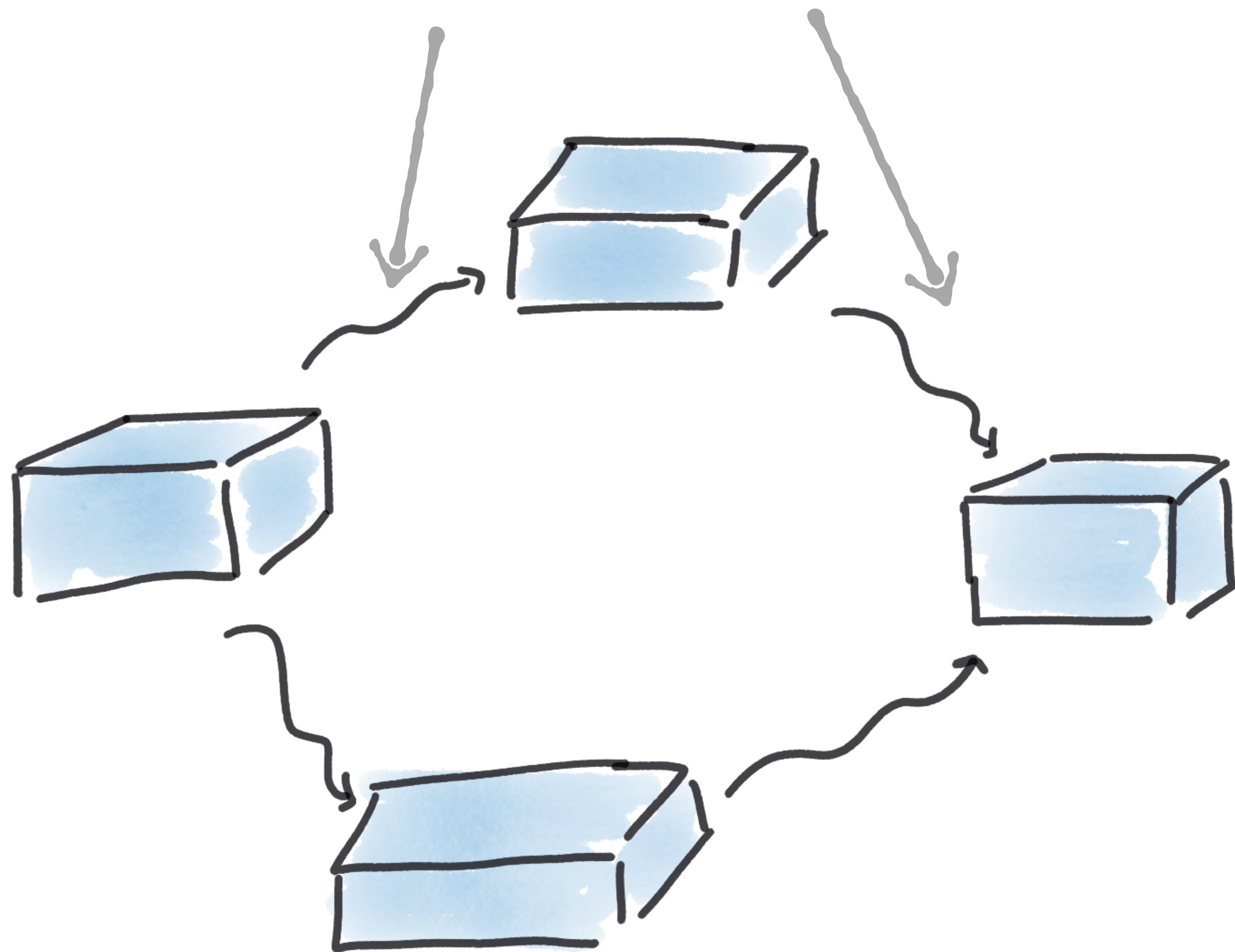
no compile-time checking

... or guaranteed function execution

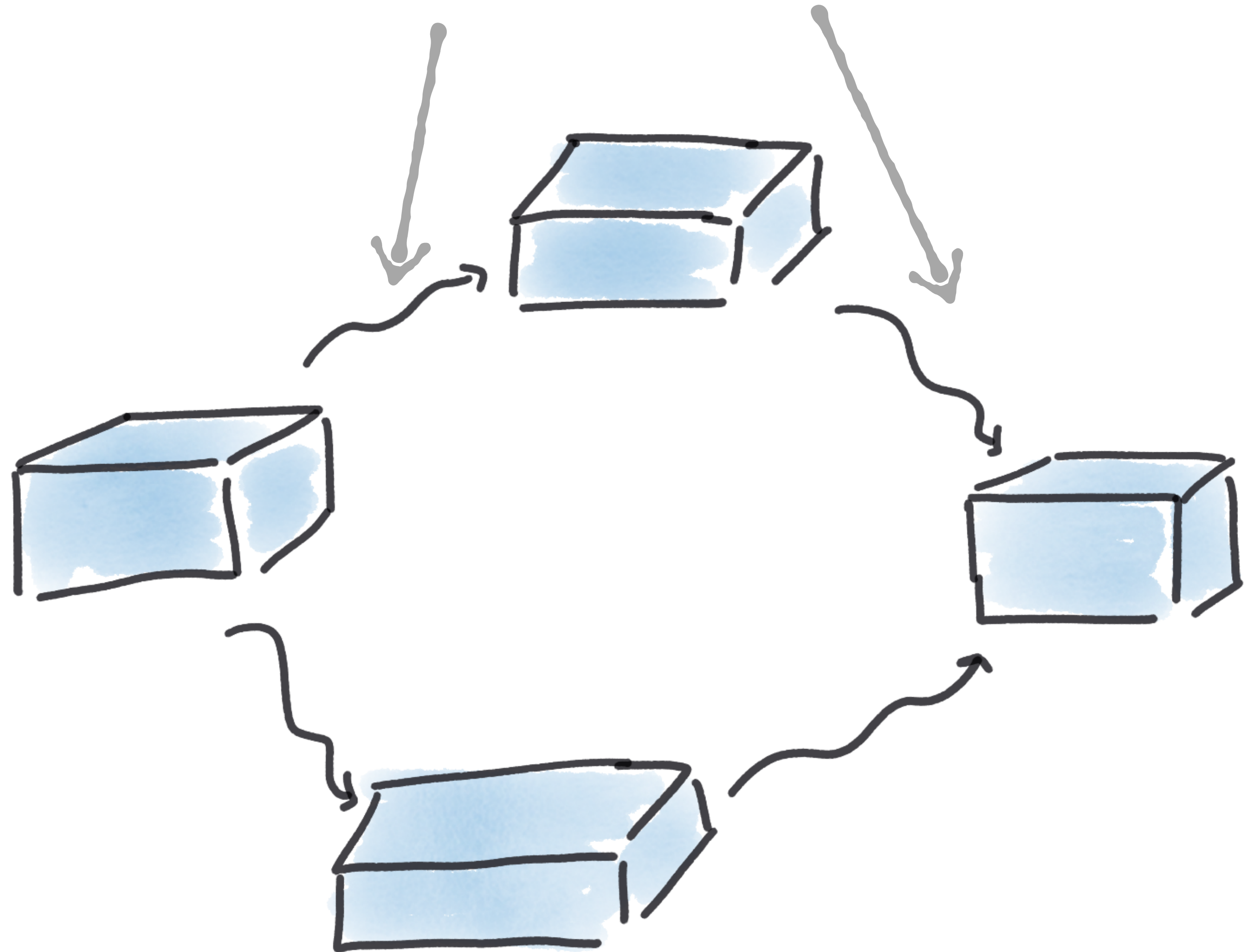
microservices  
are not the **goal**

microservices  
are not the **goal**  
they are the **means**

microservices  
are not the **goal**  
they are **a** means

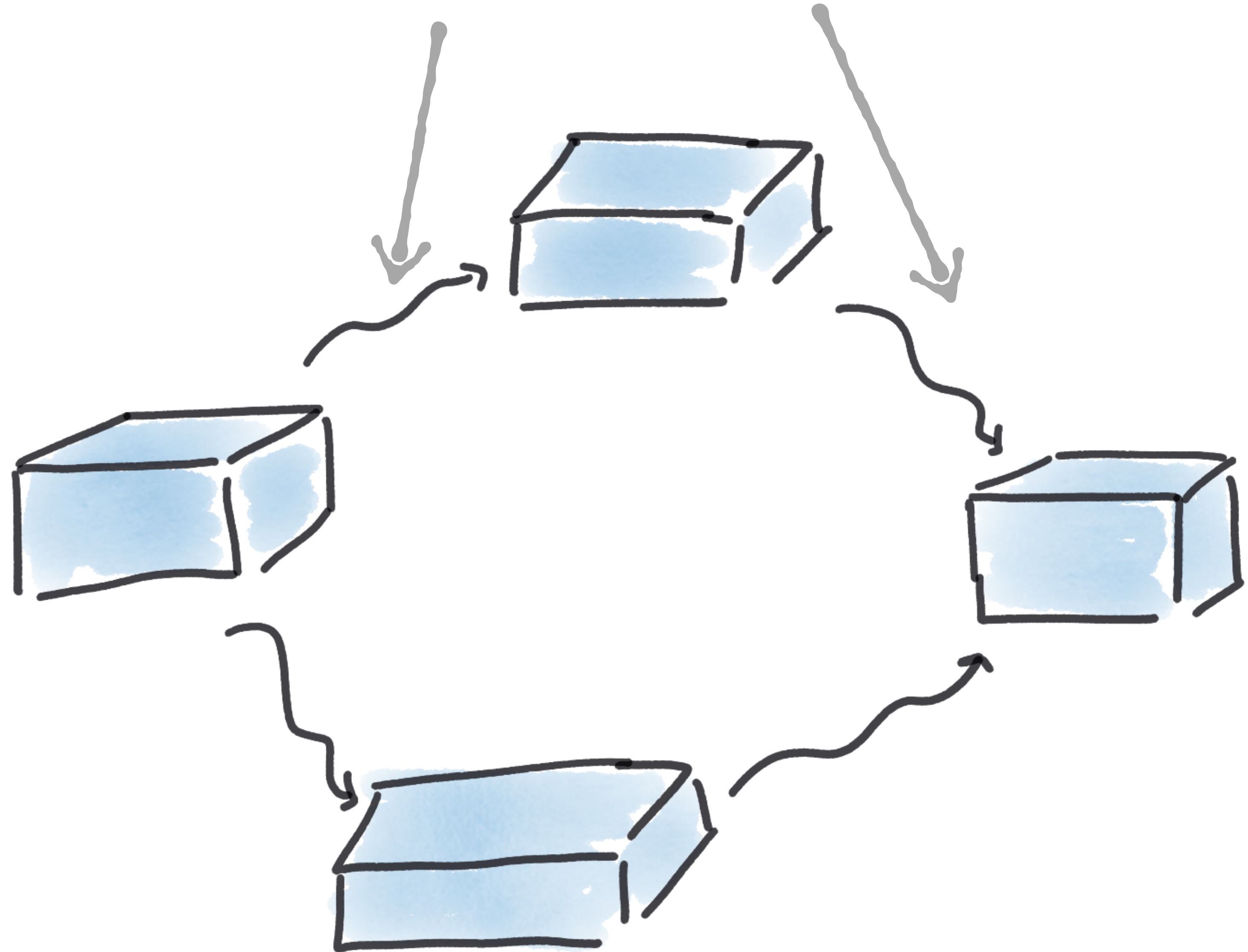


“the cloud native-ness comes from these network connections”



“the cloud native-ness comes from these network connections”

(no, it doesn't!)



# reasons **not** to do microservices

small team

not planning to release independently

don't want complexity of a service mesh - or  
worse yet, rolling your own

domain model doesn't split nicely



# Distributed computing fallacies

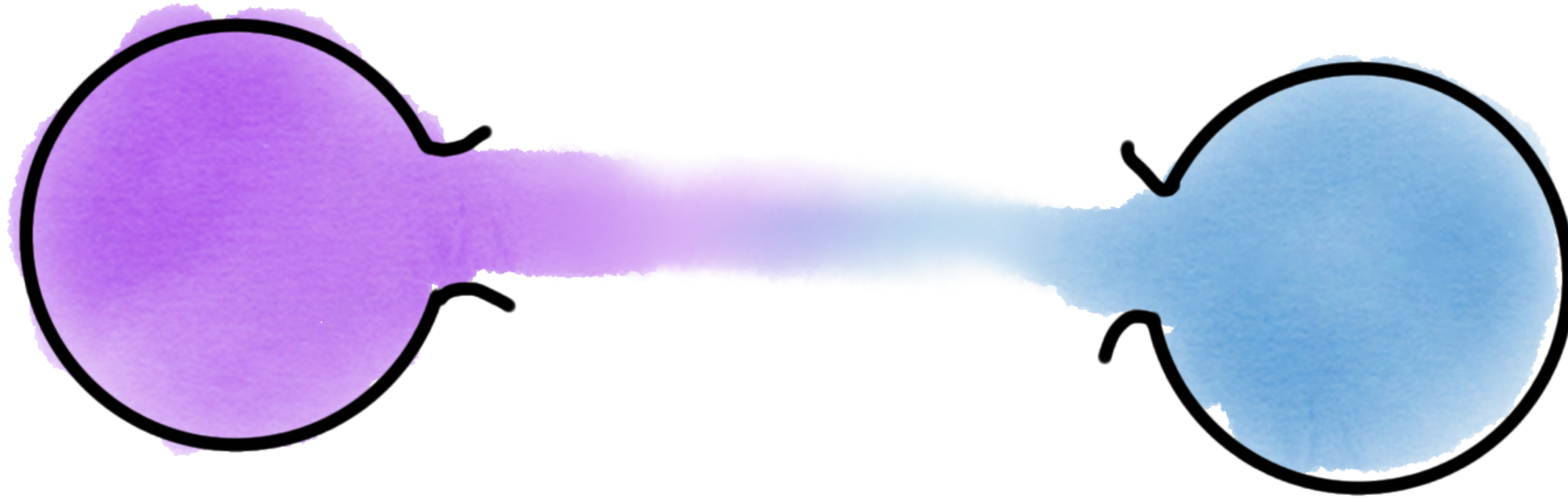
# **Distributed computing fallacies**

1. The network is reliable;
2. Latency is zero;
3. Bandwidth is infinite;
4. The network is secure;
5. Topology doesn't change;
6. There is one administrator;
7. Transport cost is zero;
8. The network is homogeneous.

# **Distributed computing fallacies**

1. The network is reliable;
2. Latency is zero;
3. Bandwidth is infinite;
4. The network is secure;
5. Topology doesn't change;
6. There is one administrator;
7. Transport cost is zero;
8. The network is homogeneous.
9. Distributed means decoupled!

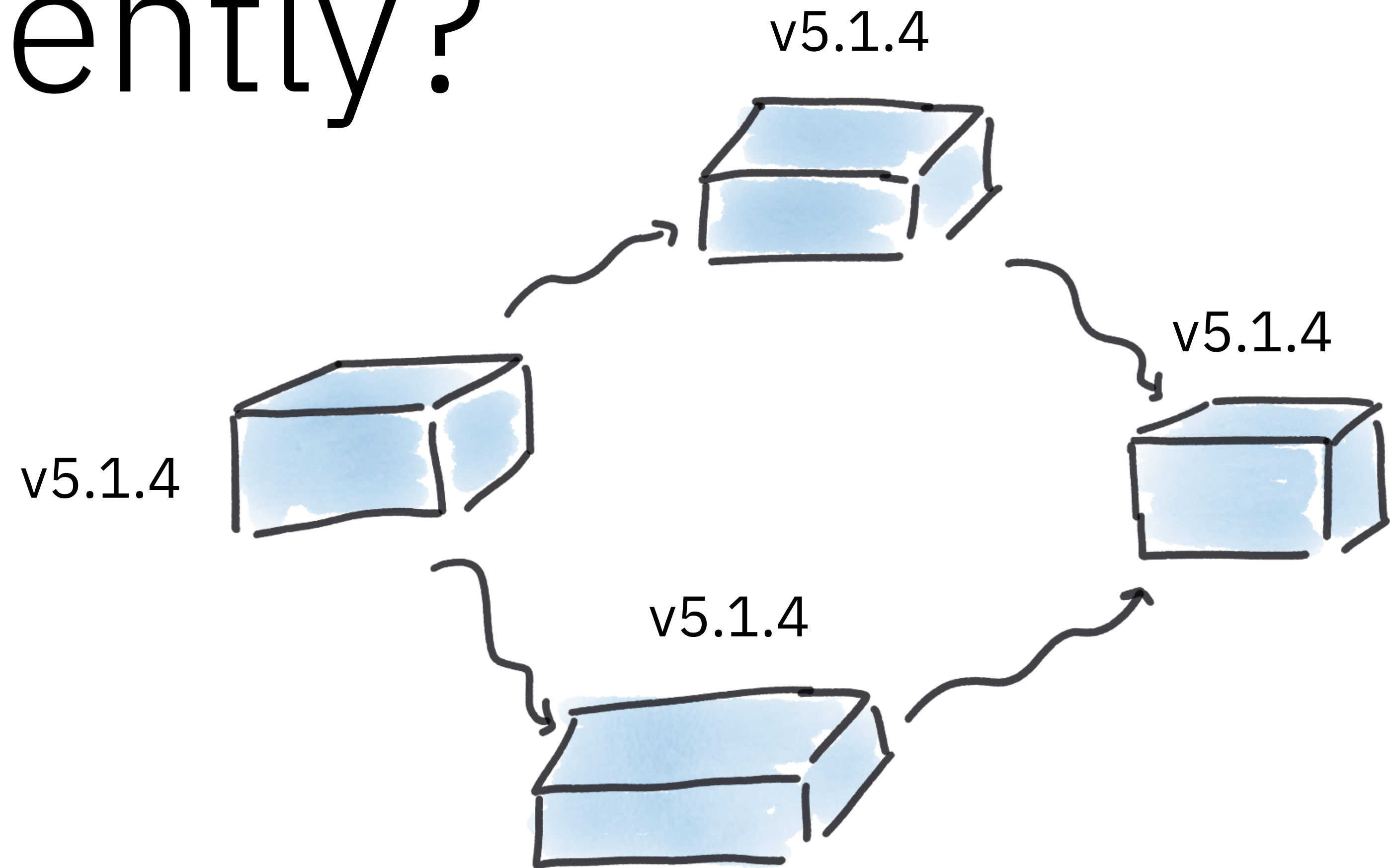
“every time we change one  
microservice, another breaks”



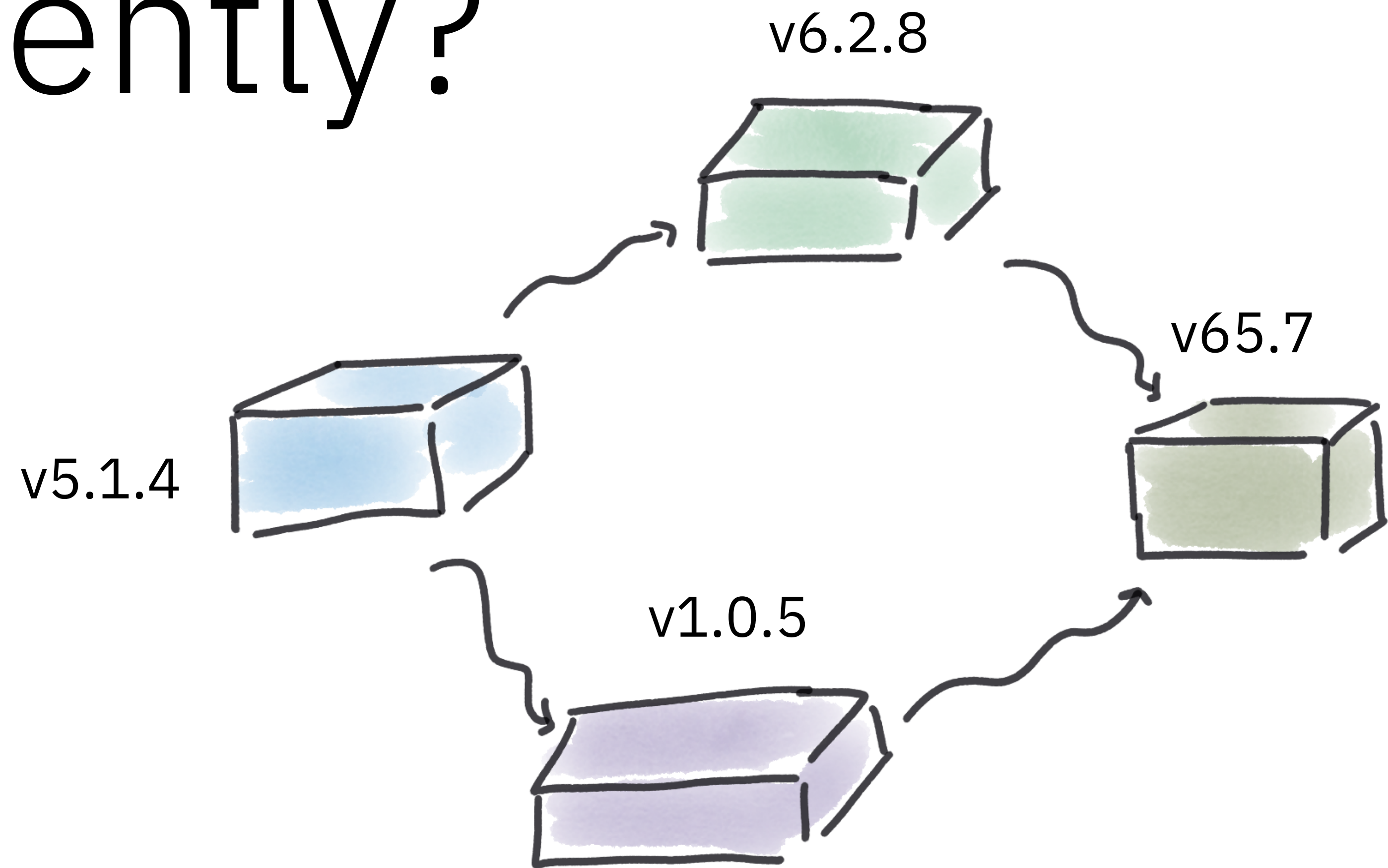
distributed != decoupled

how do you know if  
you're decoupled?

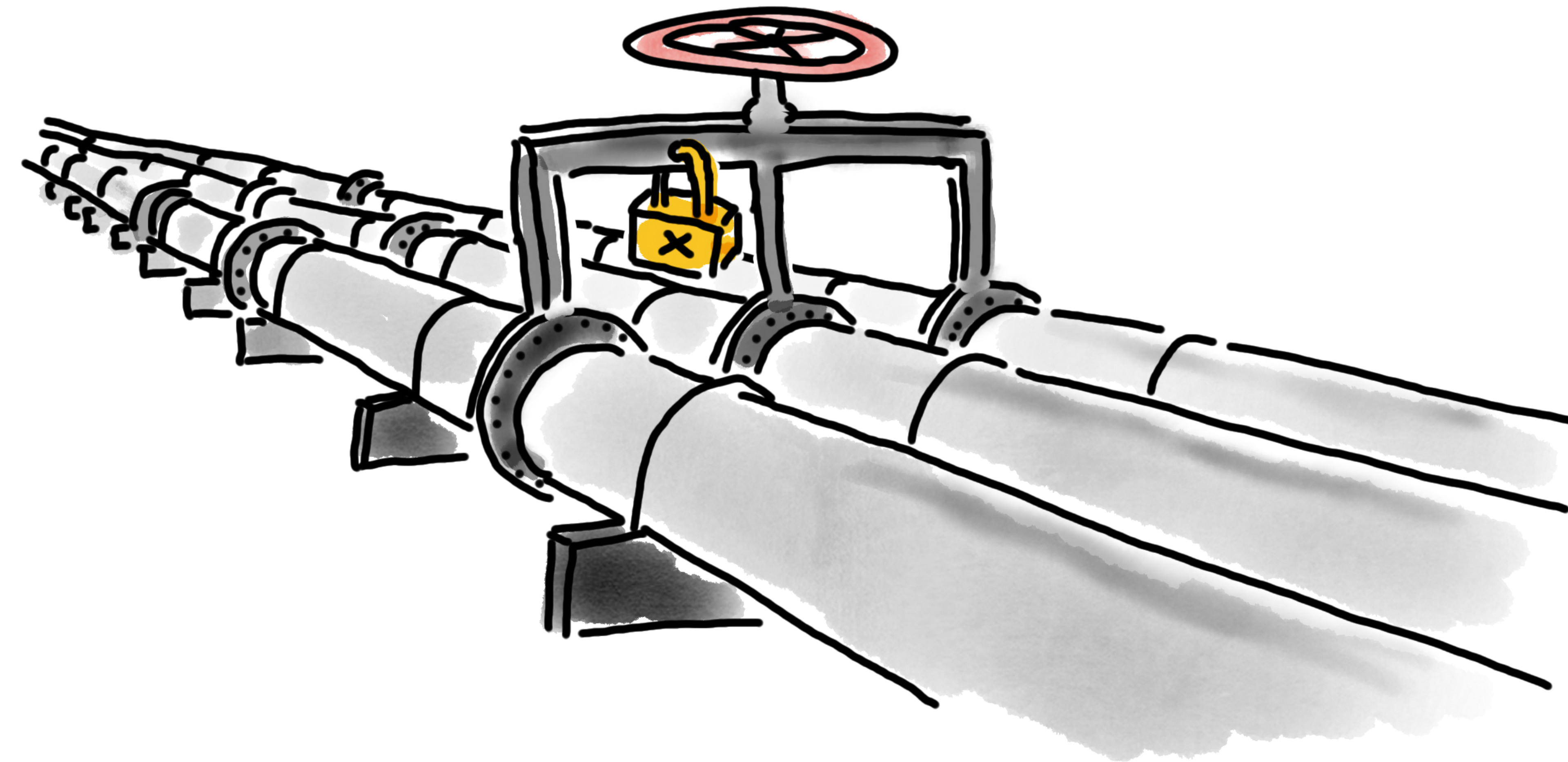
# can you deploy independently?



# can you deploy independently?



“we made our pipeline enforce releasing all our microservices at the same time.”



can you even  
deploy at all?

do **you**  
deploy?



[#684983](https://app.sli.do/event/kcvns9iw)

can you even  
release at all?

a deploy is not a release

deferred wiring

a deploy is not a release

# feature flags

**a deploy is not a release**

A/B testing

canary deploys

“we have a CI/CD”

by the way, you also  
can't buy devops

and you can't  
buy a devop

CI/CD is something you **do**  
not a tool you buy

“i’ll merge my branch  
into our CI next week”

“CI/CD ... CI/CD ... CI/CD ...  
we release every six months ...  
CI/CD ....”

# **continuous.**

I do not think that word means  
what you think it means.

“we’re going too slowly.  
we need to get rid of COBOL  
and make microservices!”

“we’re going too slowly.  
we need to get rid of COBOL  
and make microservices!”

“... but our release board  
only meets twice a year.”

“we can’t actually **release**  
a microservice without the  
release board and all the other  
microservices need to be  
released at the same time and  
the manual QA cycle and the  
release checklist and the  
incantations and the death march and  
the sacrificing a goat and the spreadsheet  
and the moon being in the right phase and...”



“we can’t ship until we  
have more confidence  
in the quality”



“our tests aren’t  
automated”

“we don’t know if  
our system works”

if you're going to be spaghetti,  
at least be **tested** spaghetti

“we can’t release because we  
don’t seem to be able to  
actually change things”



why change is hard:

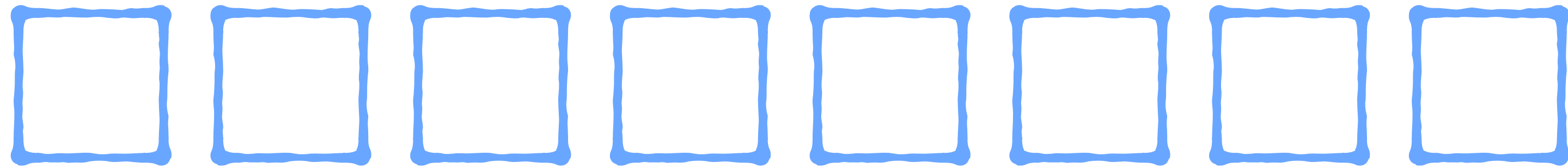


the enterprise hairball

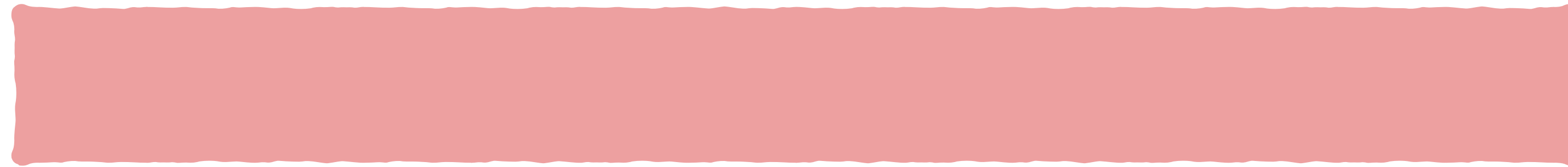
**our layer**  
("business logic")



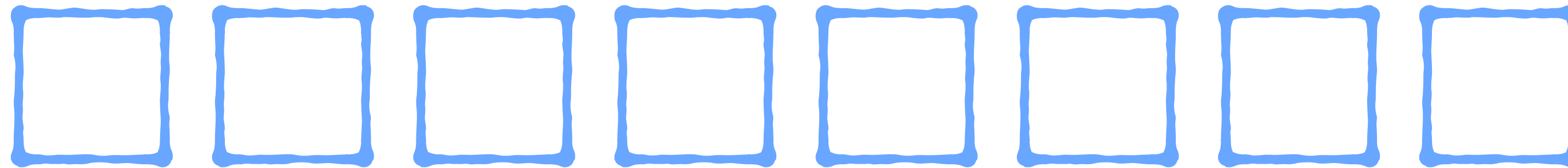
**our layer**  
("business logic")



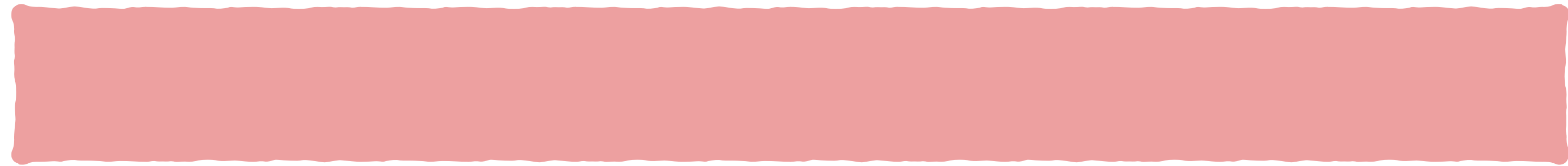
front-end layer



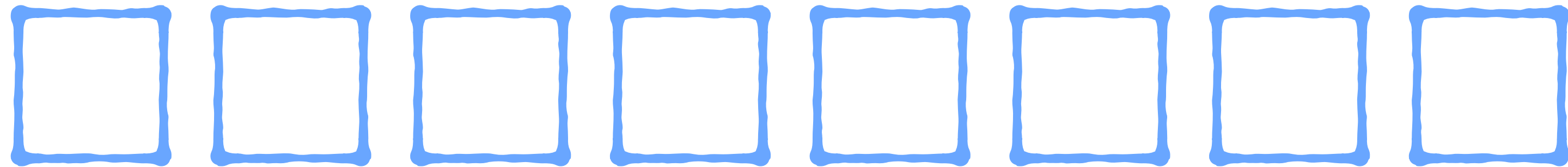
our layer  
("business logic")



front-end layer



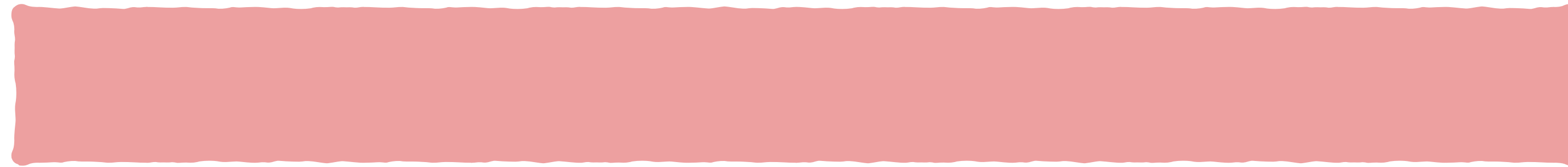
our layer  
("business logic")



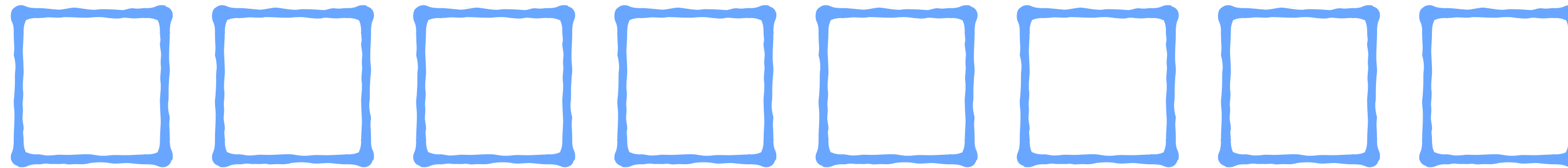
database layer



front-end layer



our layer  
("business logic")



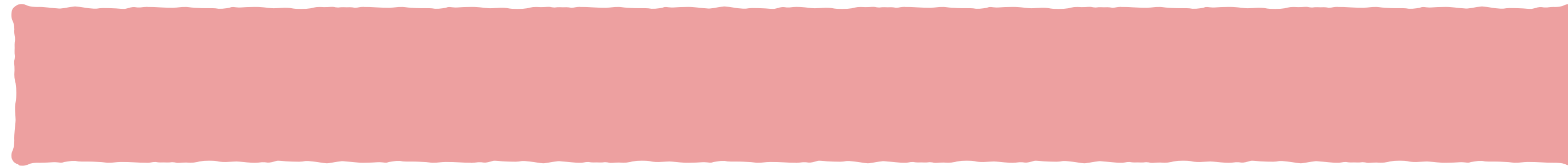
integration layer  
("panicked sandwich")



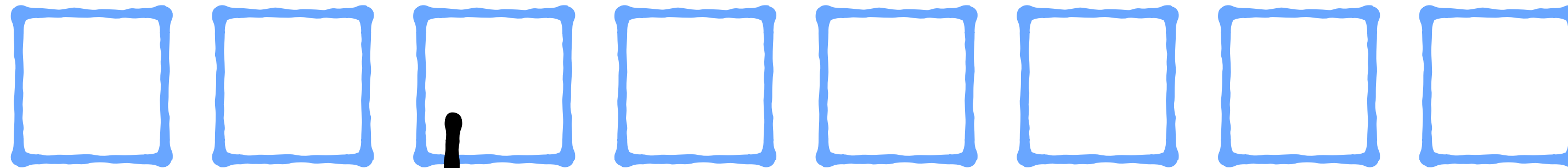
database layer



front-end layer



our layer  
("business logic")



integration layer  
("panicked sandwich")



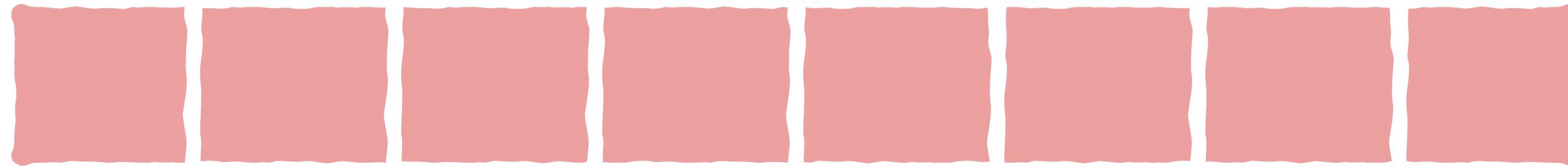
database layer



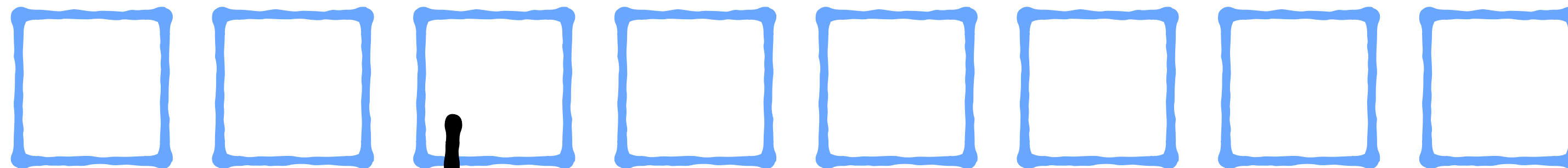
oh wait ...  
every change needs a  
front-end change too



front-end layer



our layer  
("business logic")



integration layer  
("panicked sandwich")

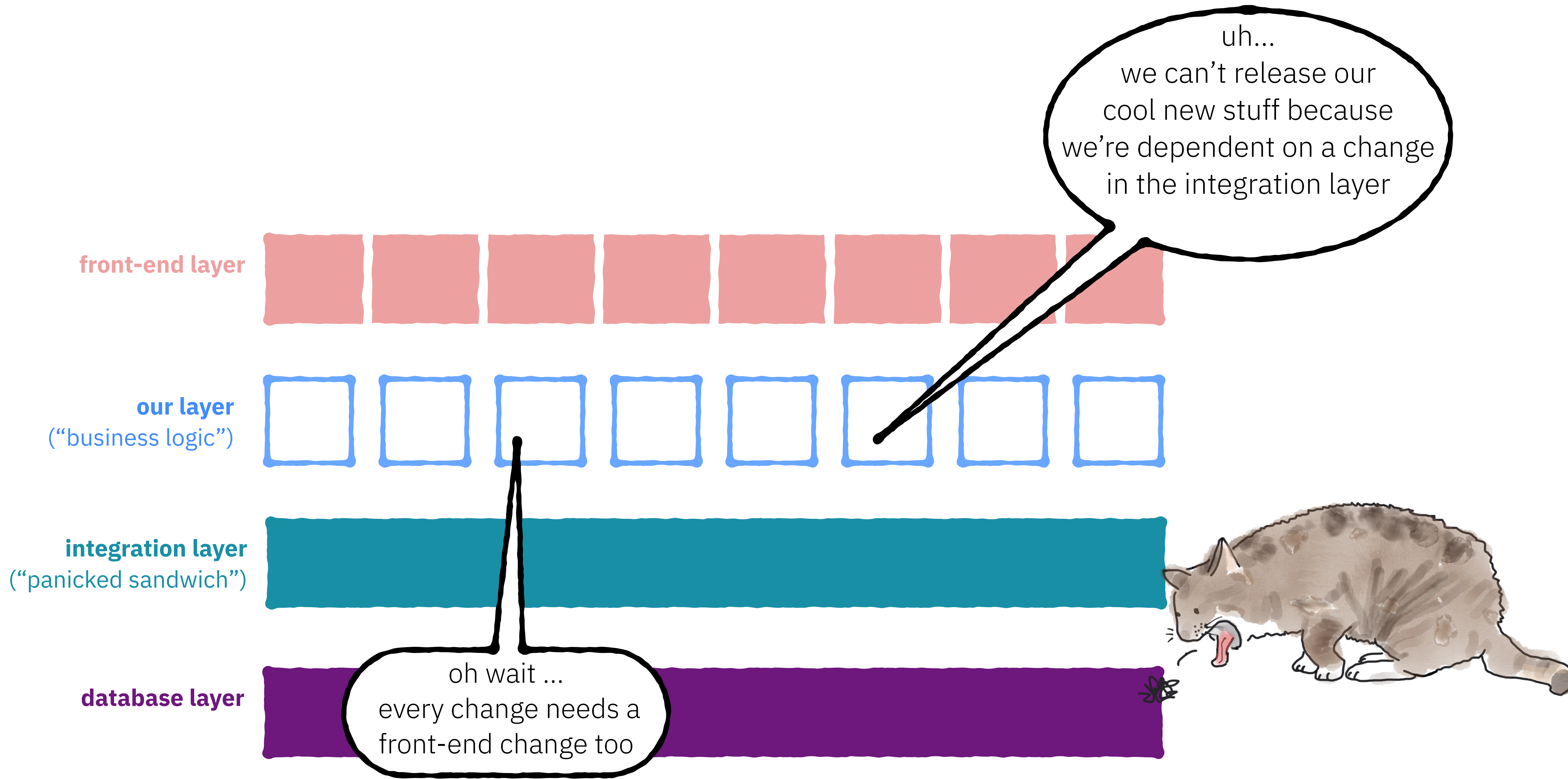


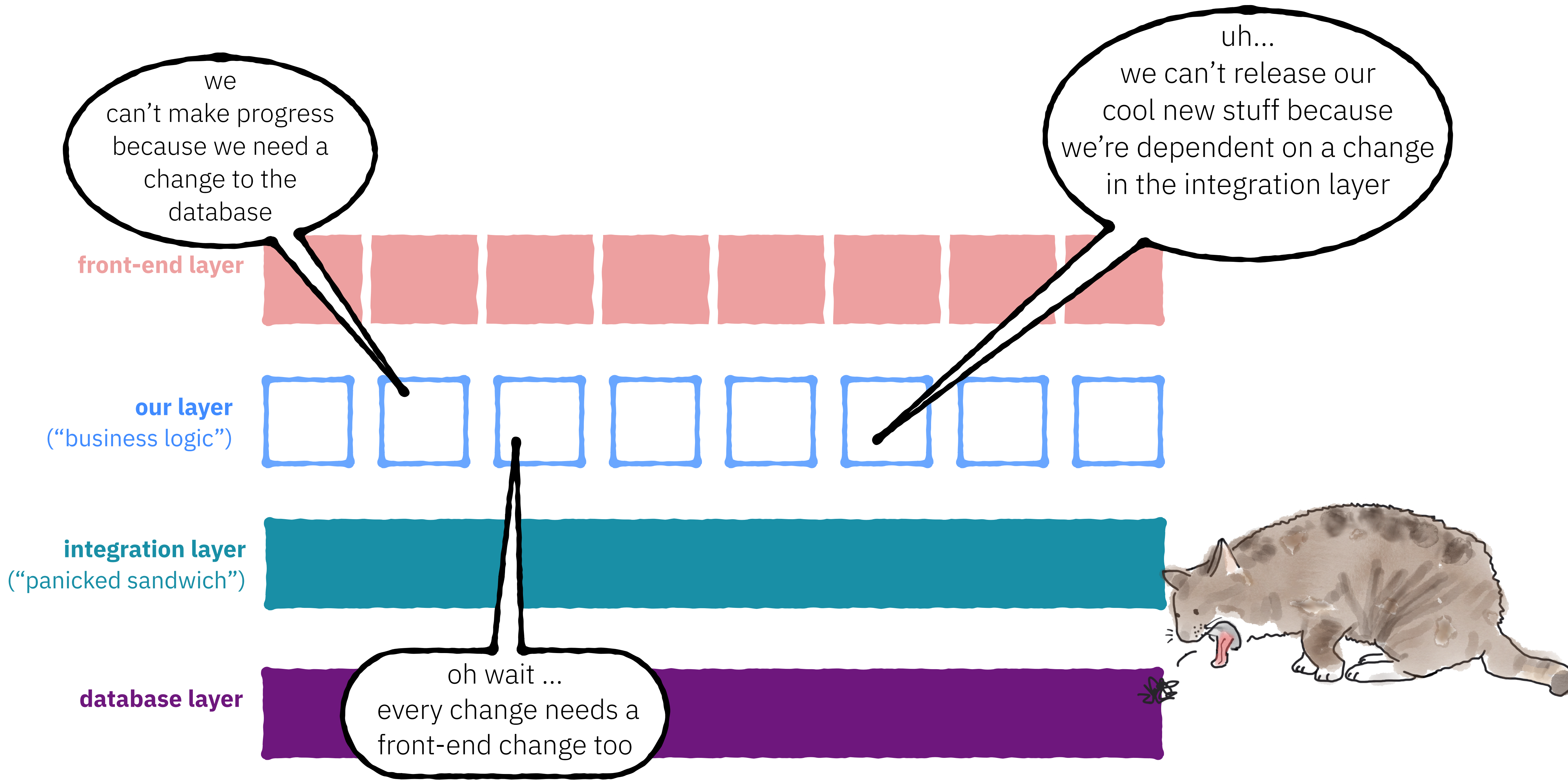
database layer

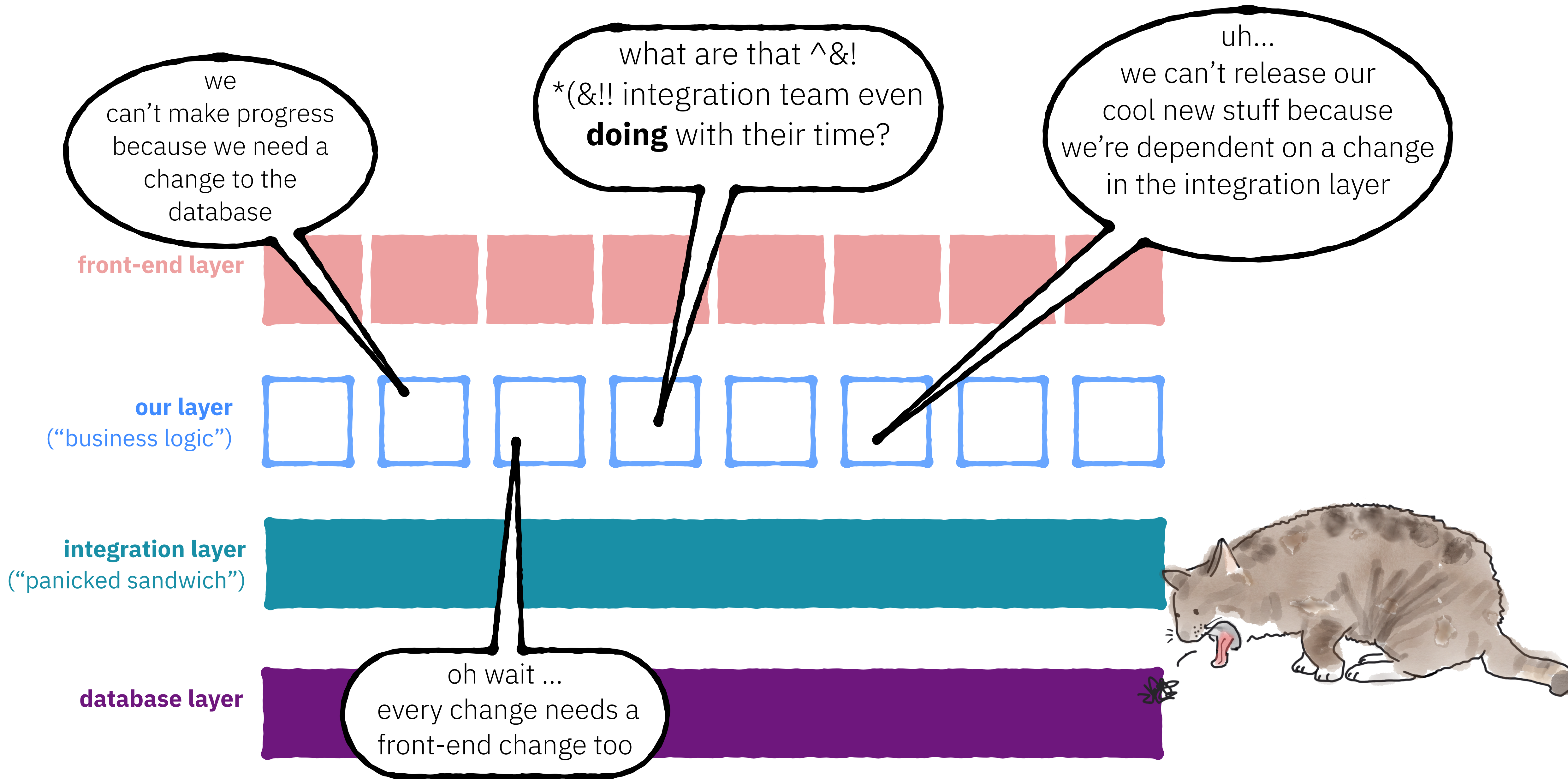


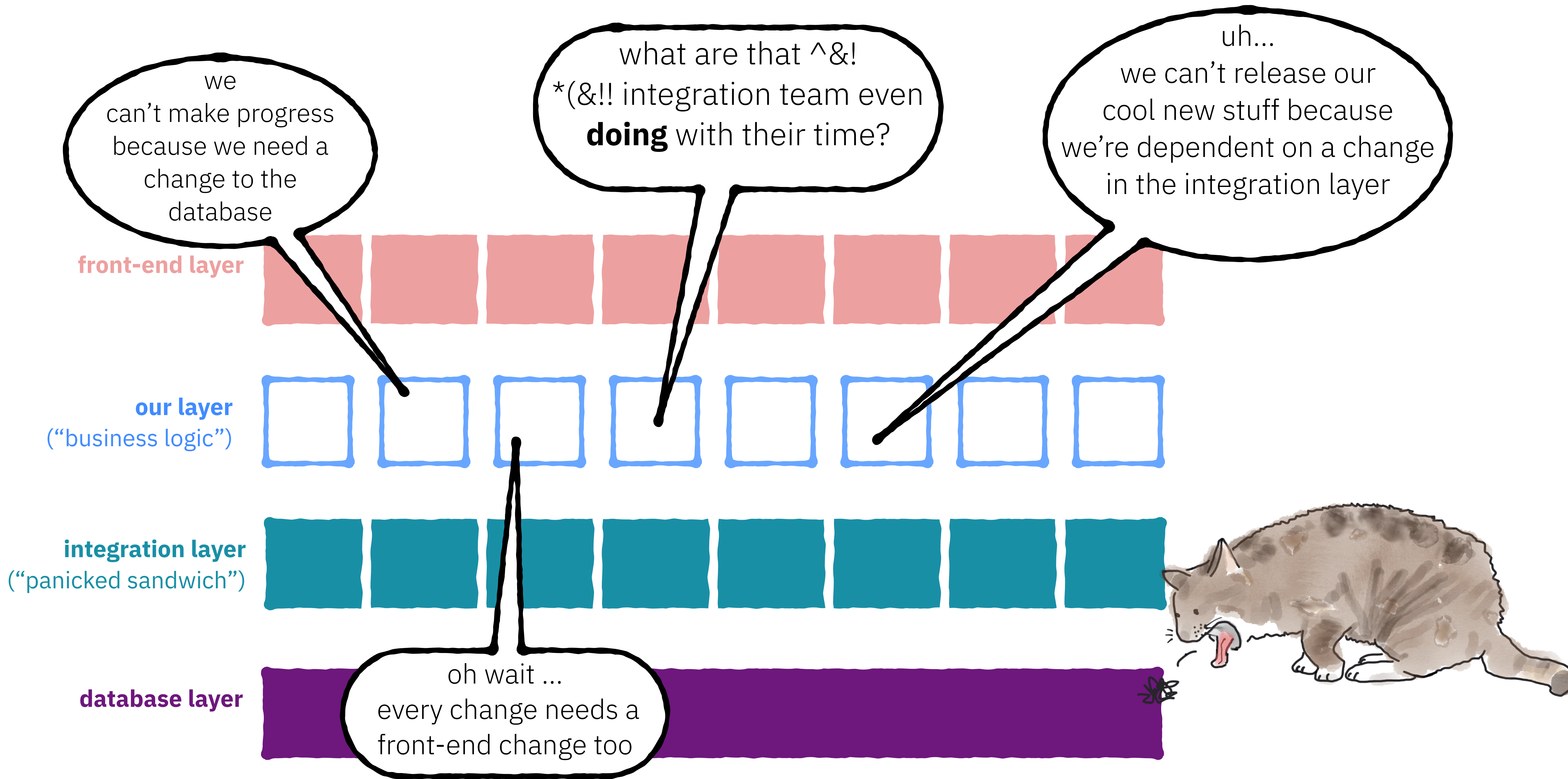
oh wait ...  
every change needs a  
front-end change too

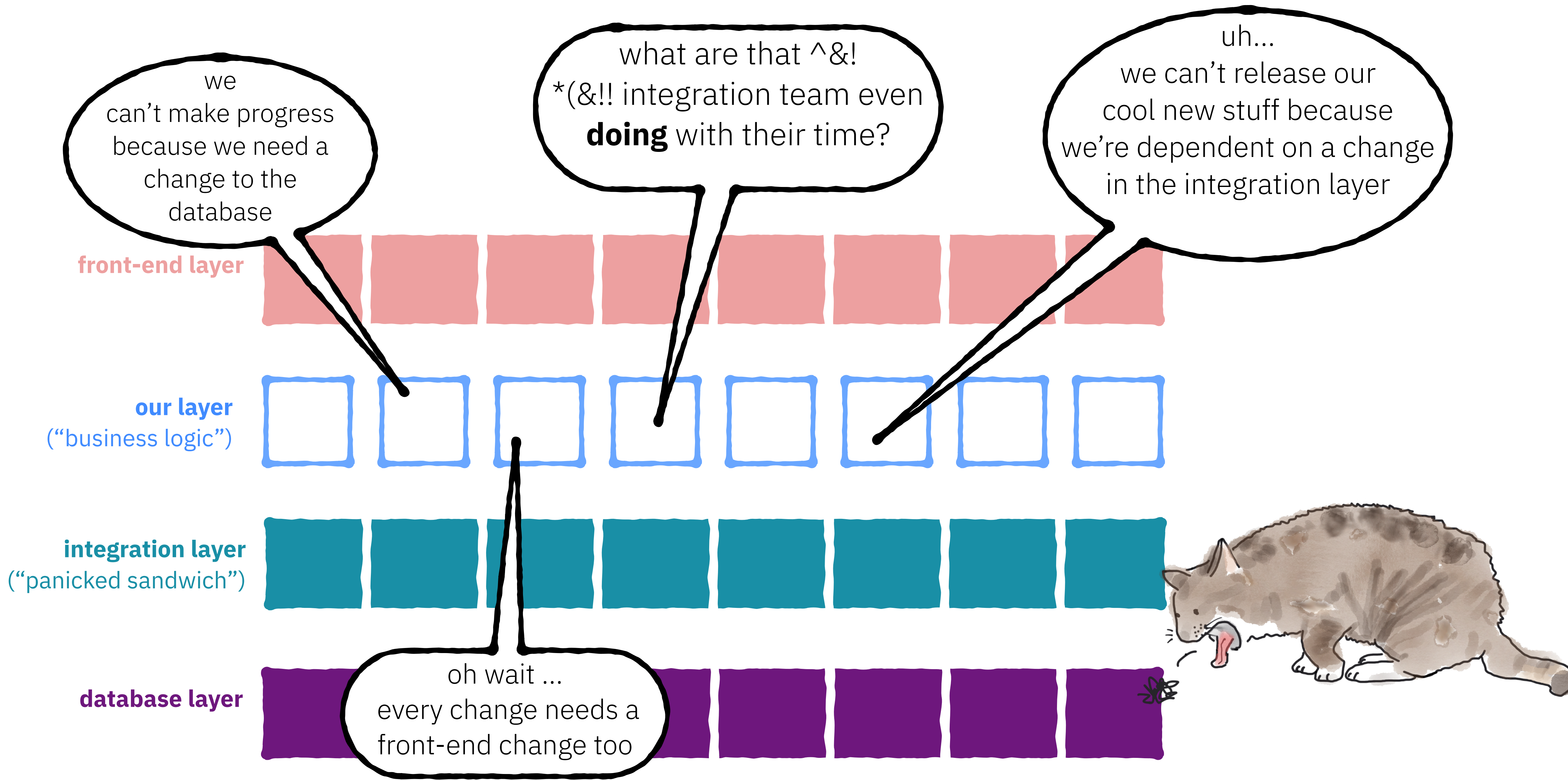












a new approach to ops

# SRE

a new approach to risk

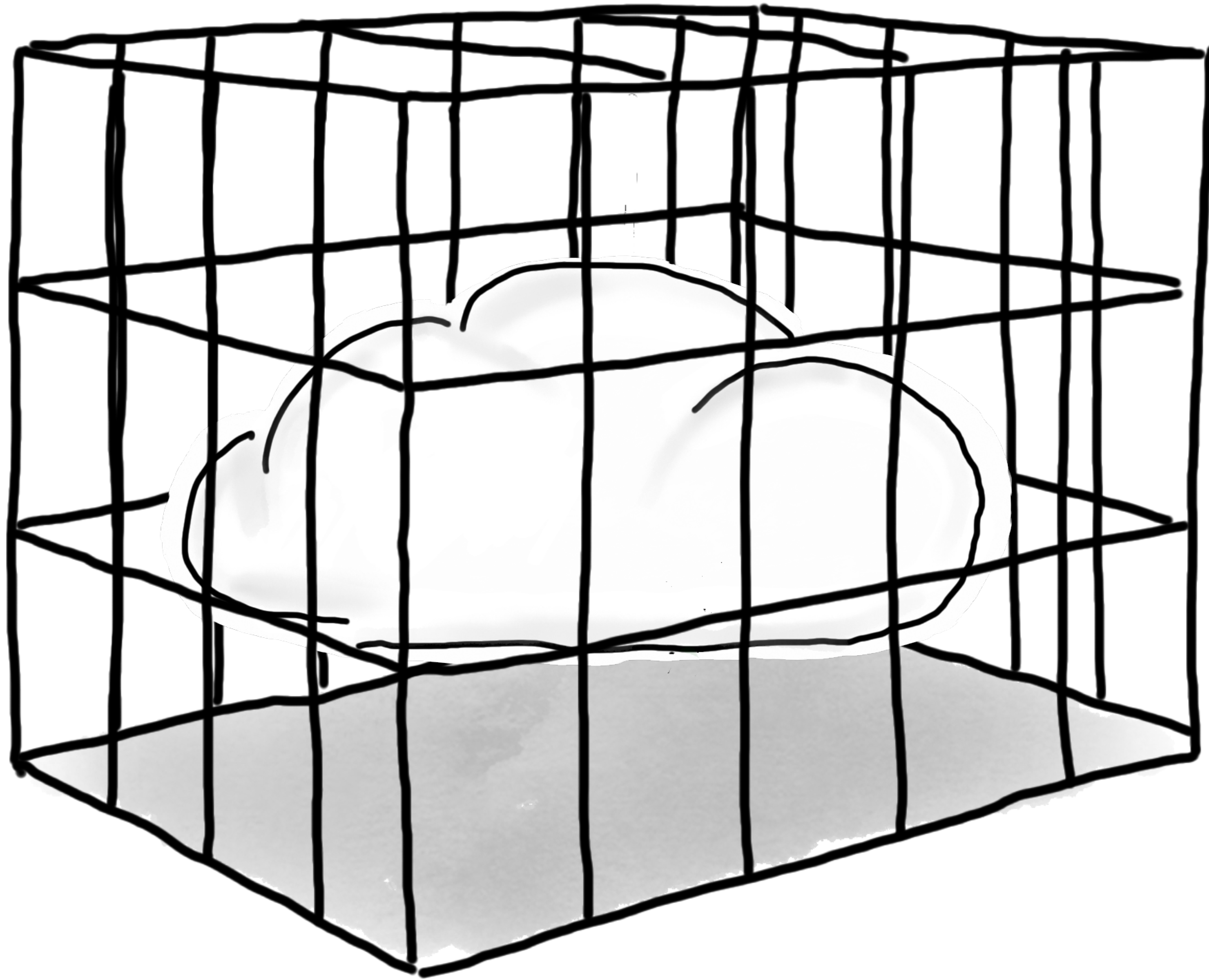
de-risk through  
automation, not paperwork



#WTFIsCloudNative

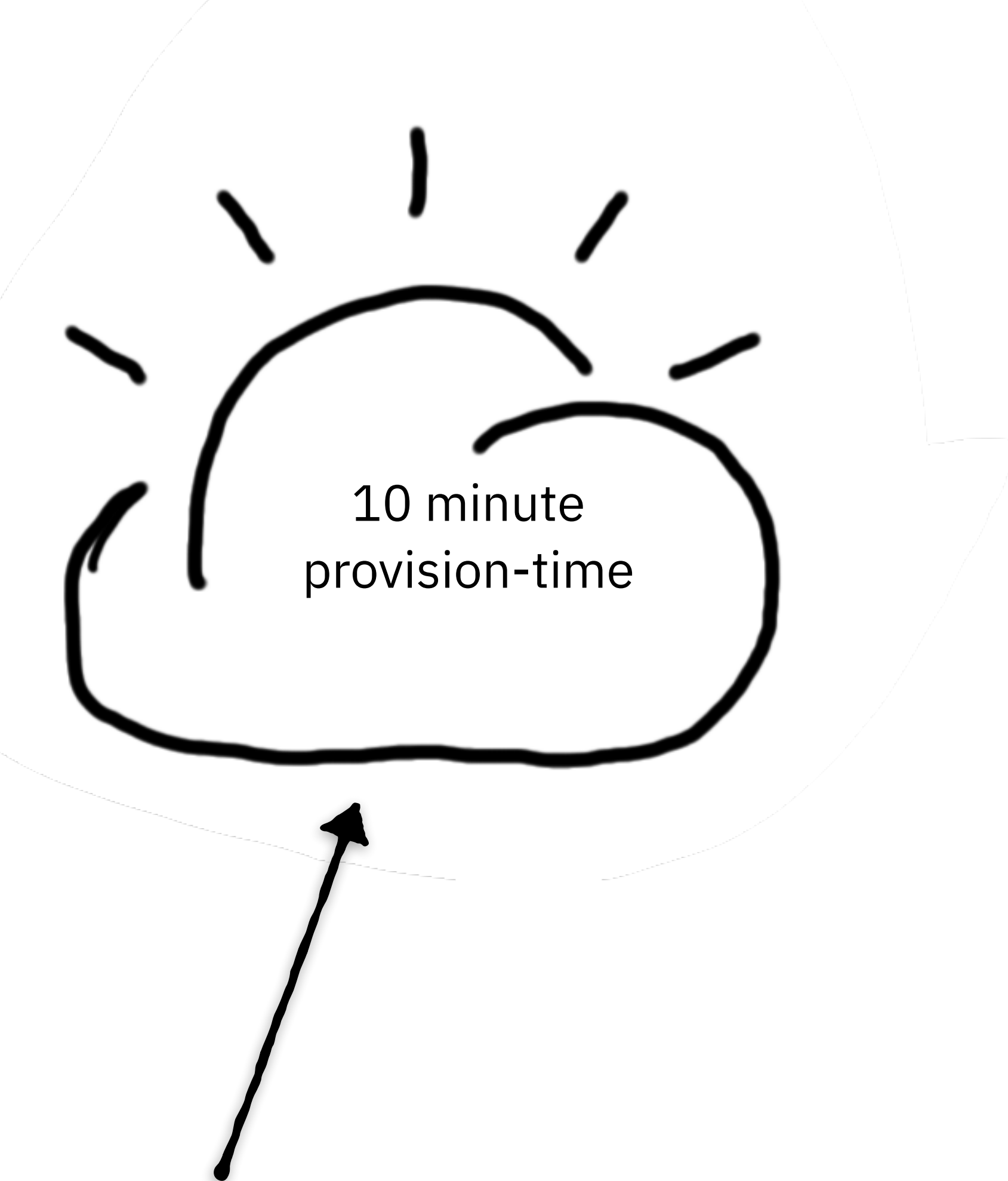
@holly\_cummins

#IBM



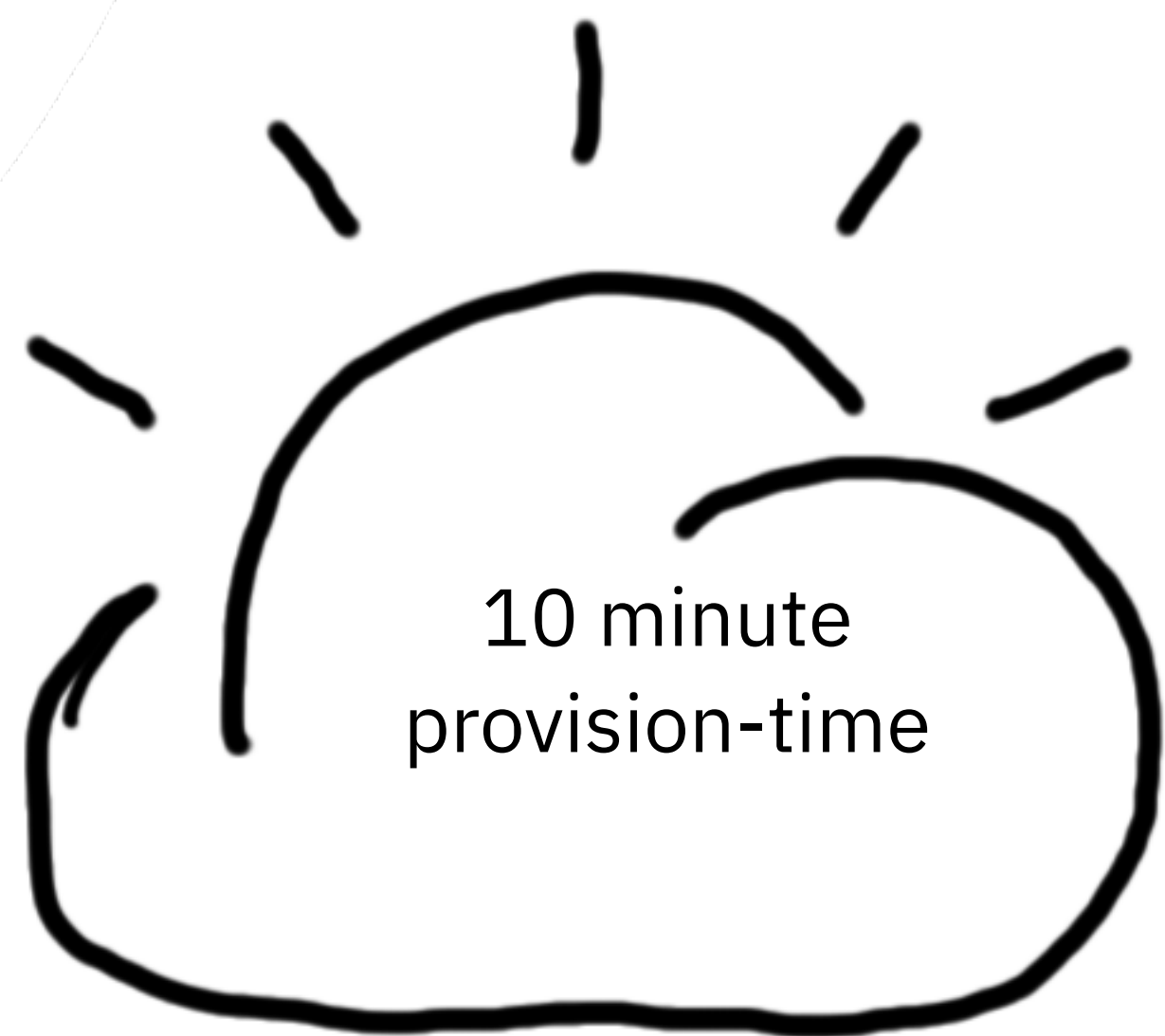
old-style governance  
isn't going to work

“this provisioning  
software is broken”



what we sold

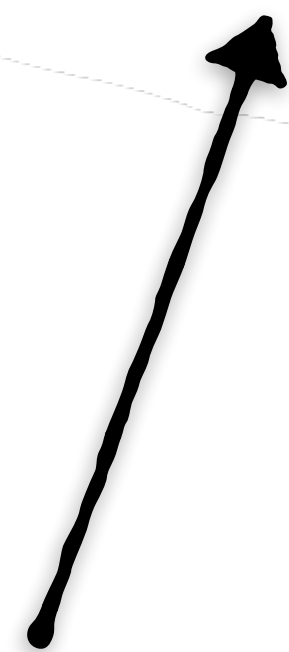
“this provisioning  
software is broken”



what the  
client  
thought  
they'd got

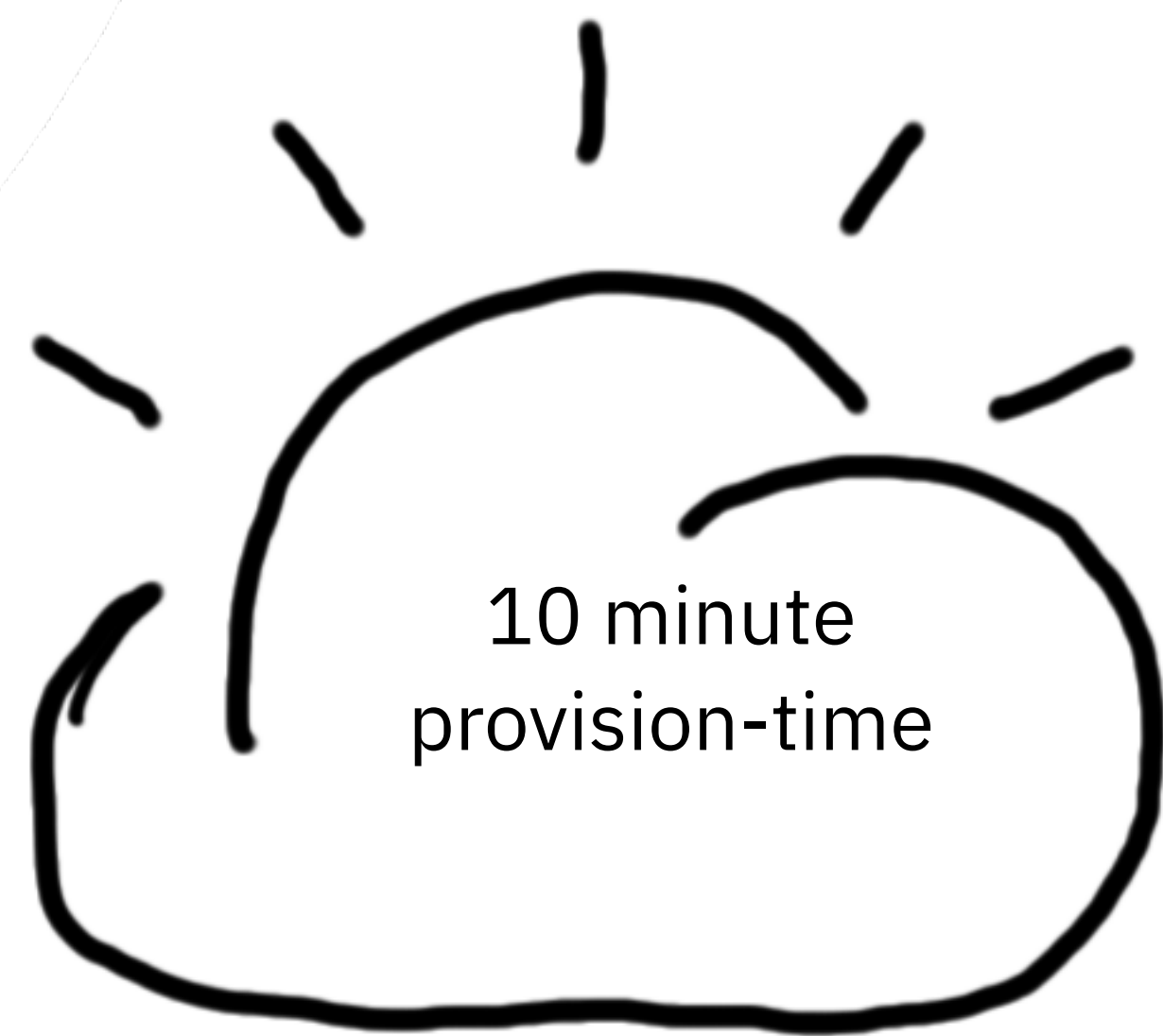


3 month  
provision-  
time



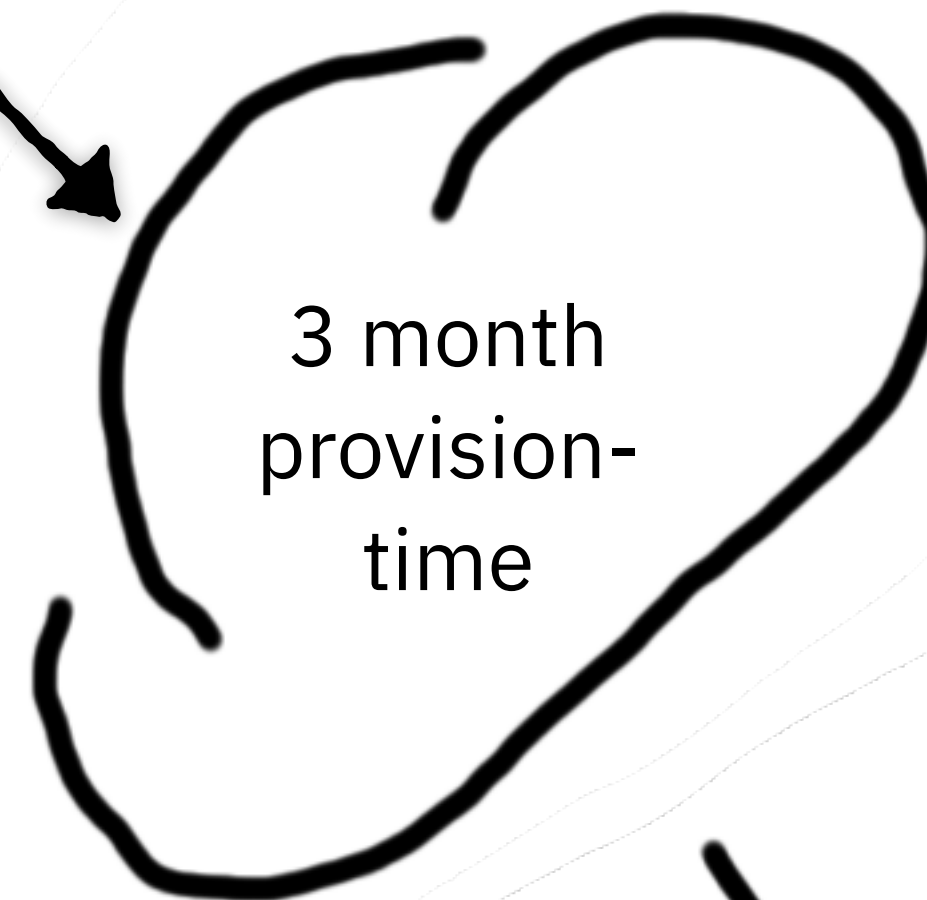
what we sold

“this provisioning  
software is broken”



what we sold

what the client thought they'd got

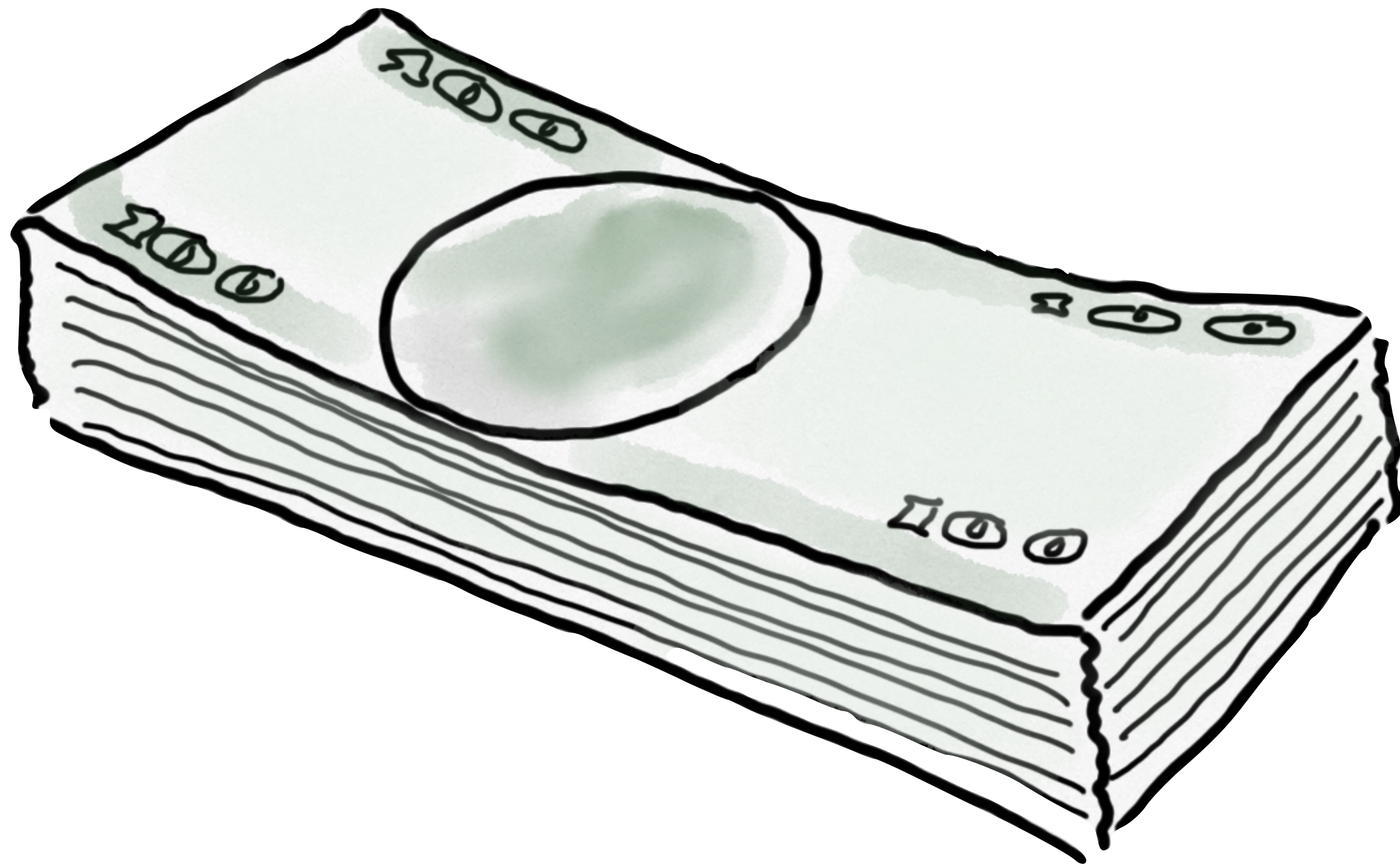


the reason



“this provisioning software is broken”

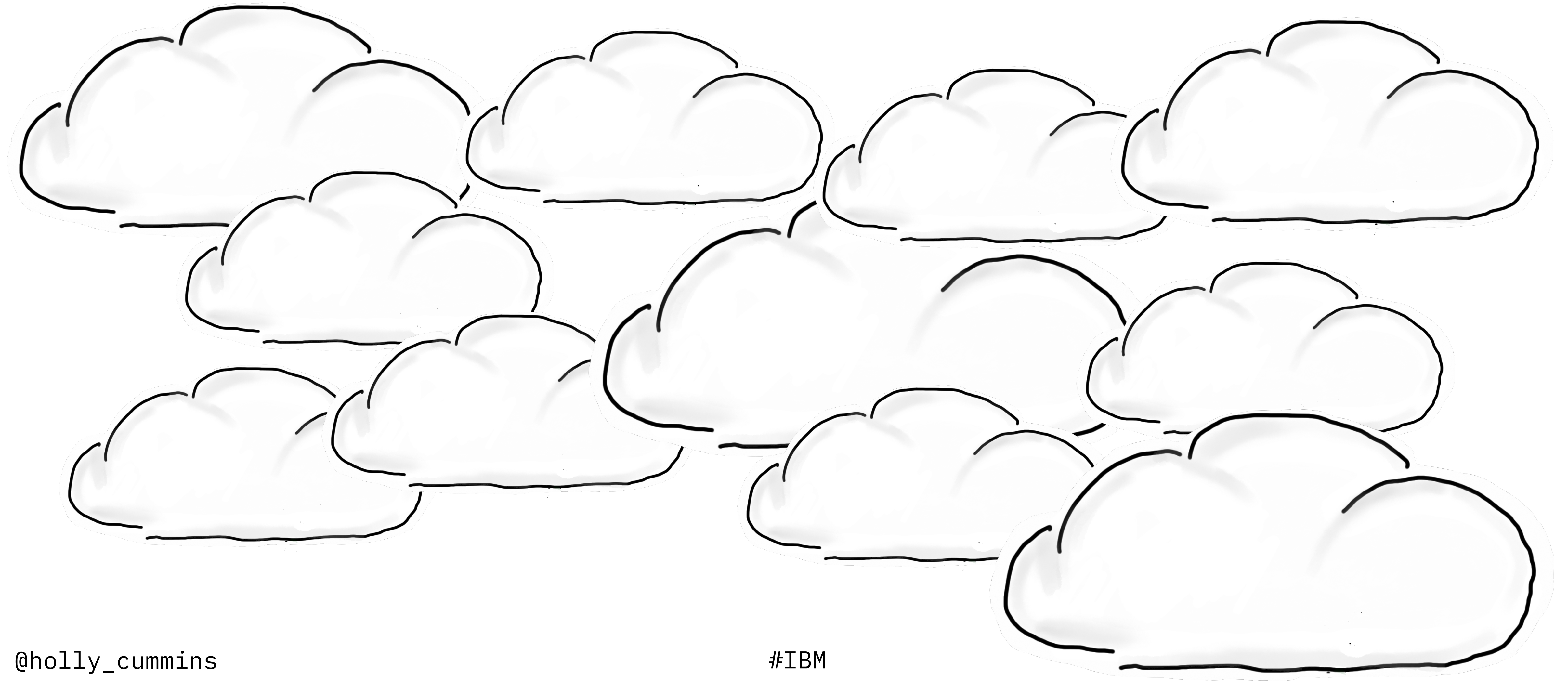
# one thing you **can** buy



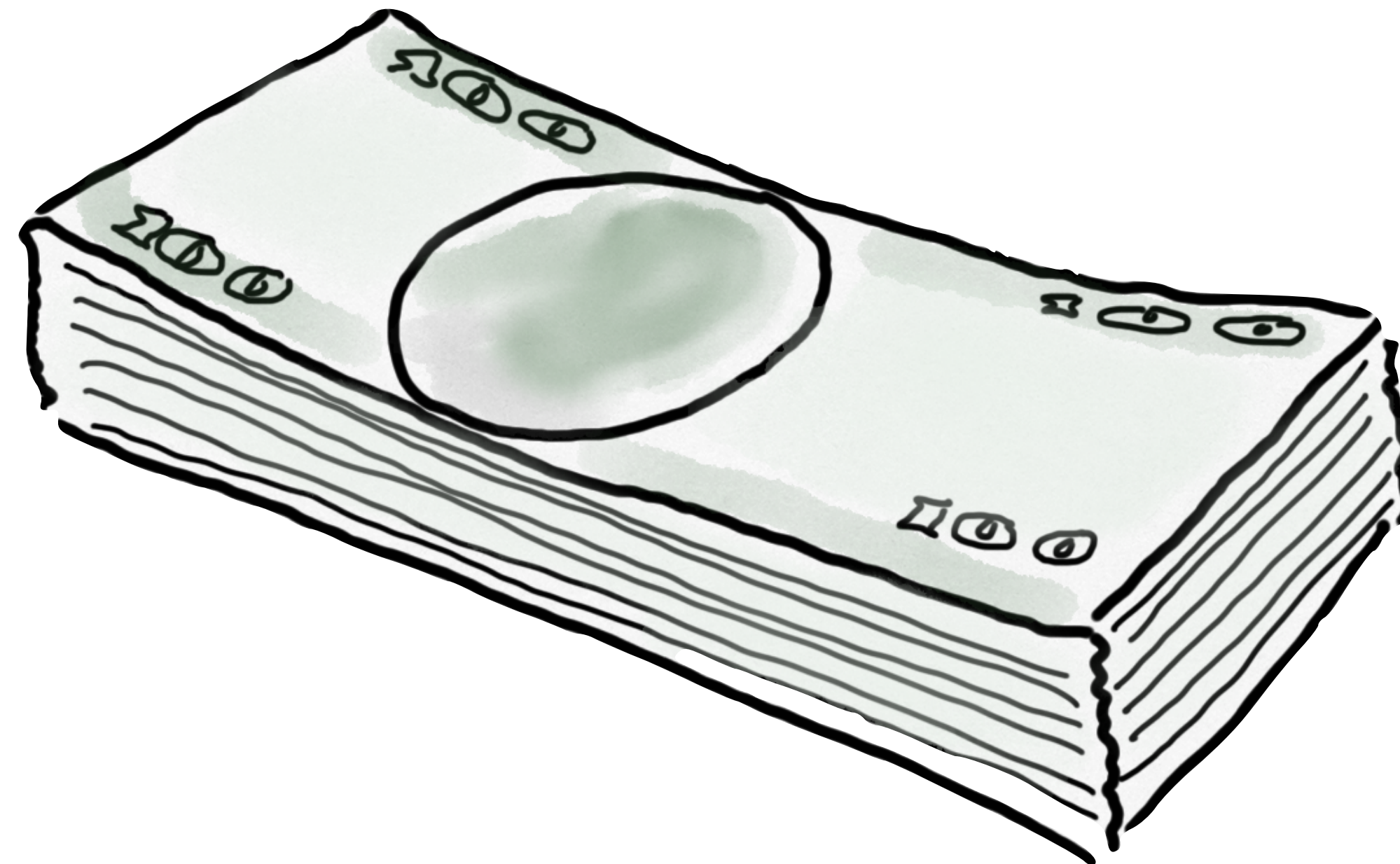
# one thing you **can** buy



# one thing you **can** buy



you may not want all  
the cloud you bought



you may not want all  
the cloud you bought

true story, unfortunately



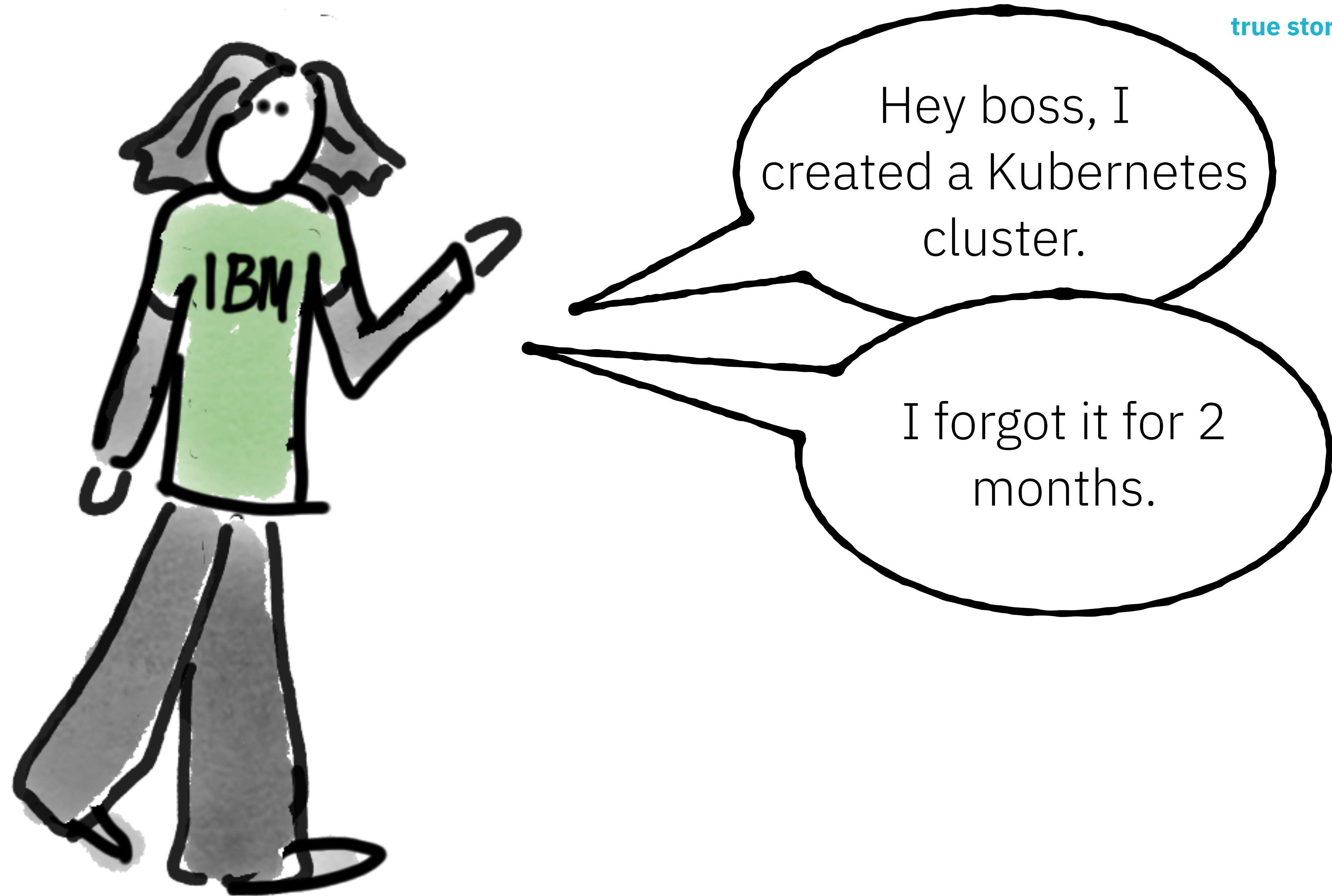
Hey boss, I  
created a Kubernetes  
cluster.

Holly

@holly\_cummins

#IBM

true story, unfortunately



Holly

@holly\_cummins

#IBM

true story, unfortunately



Hey boss, I  
created a Kubernetes  
cluster.

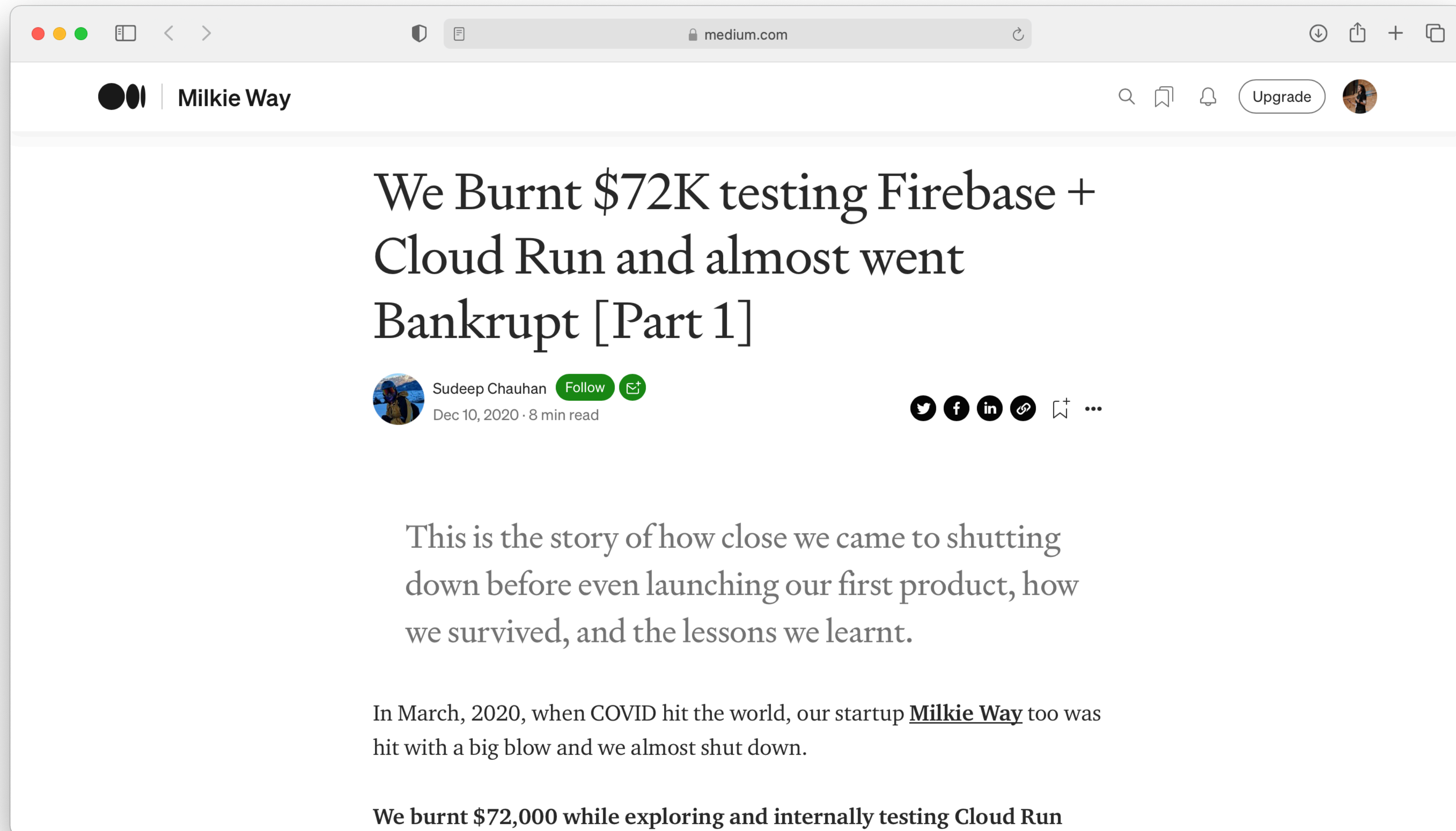
I forgot it for 2  
months.

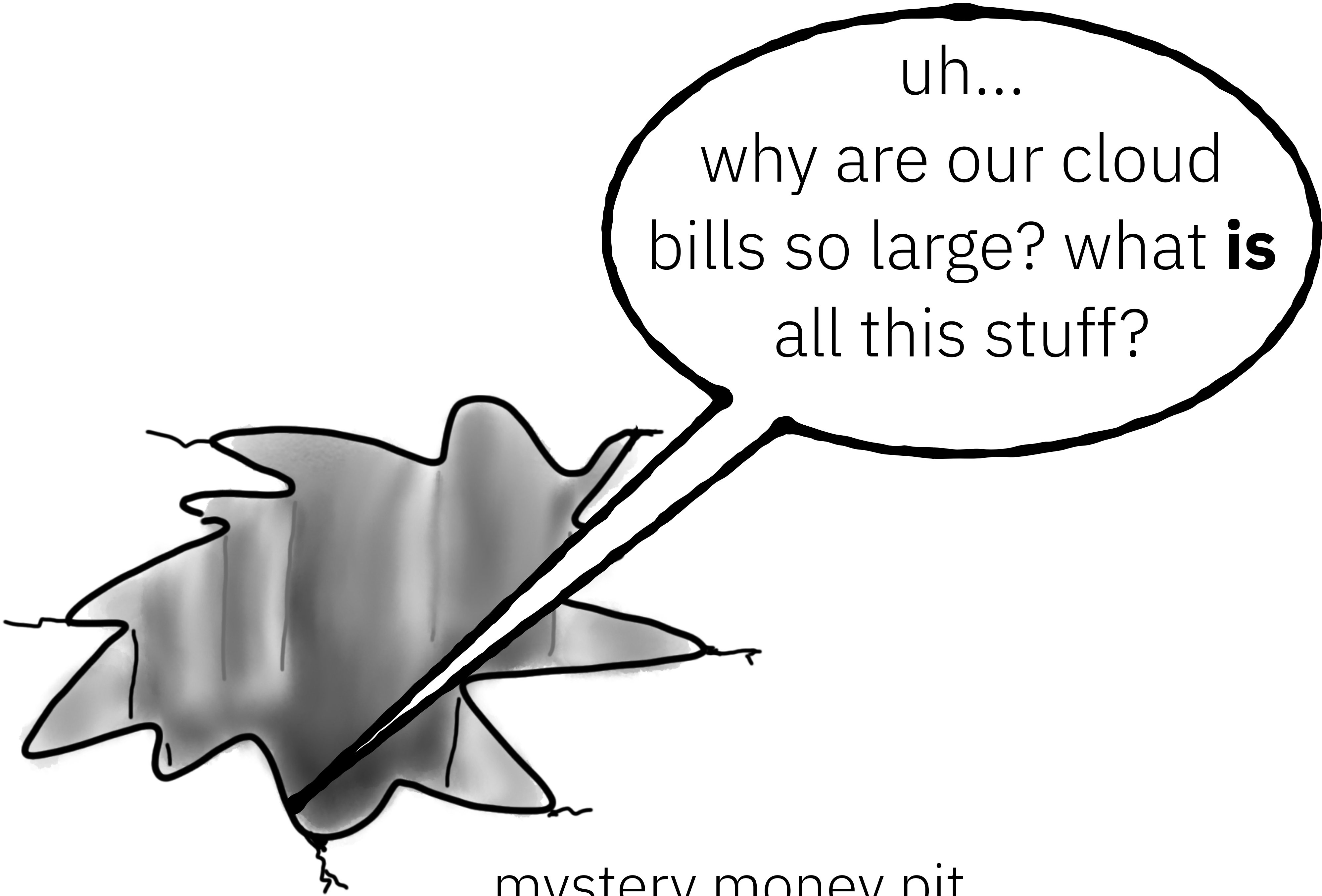
... and it's £1000 a  
month.

Holly

@holly\_cummins

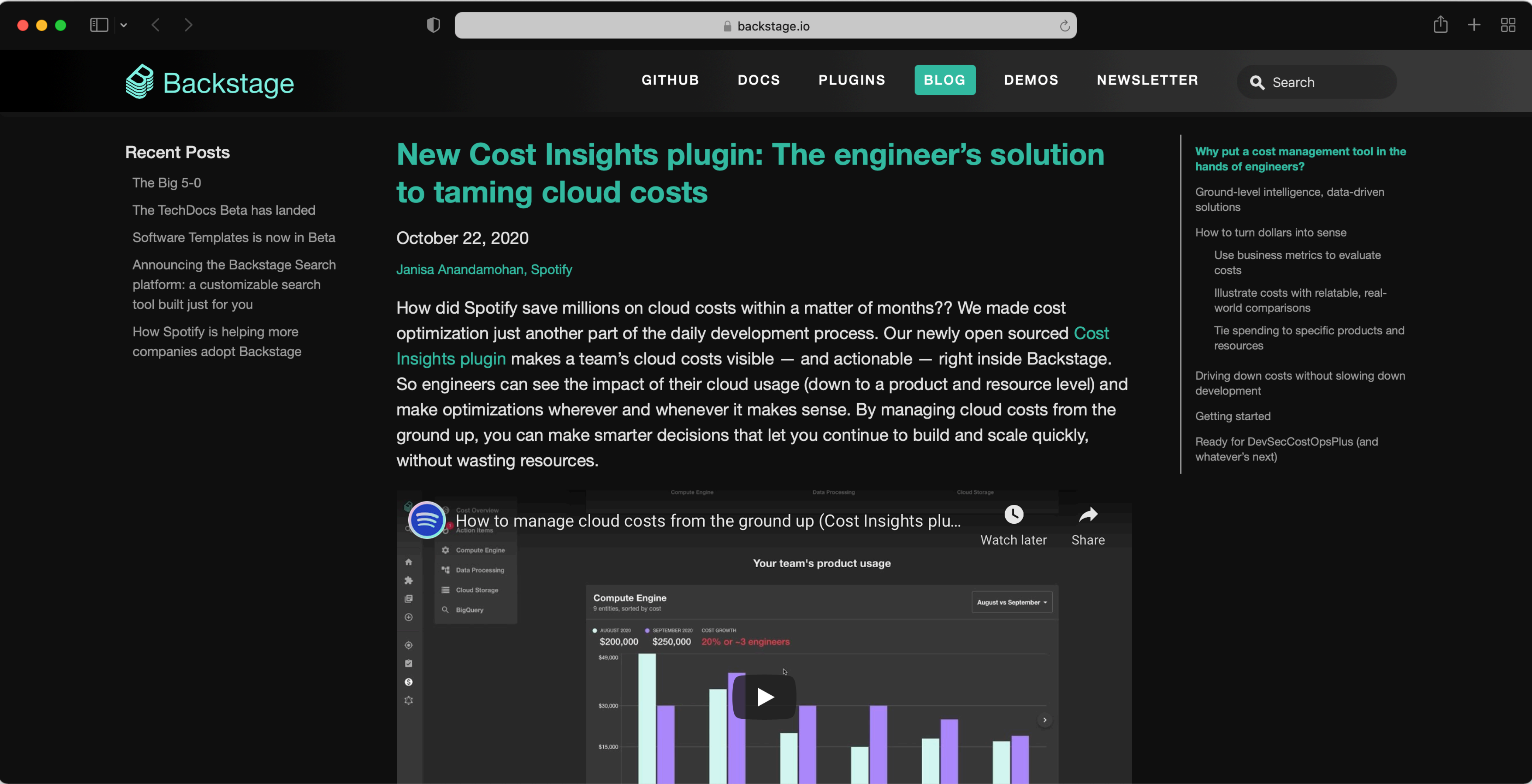
#IBM

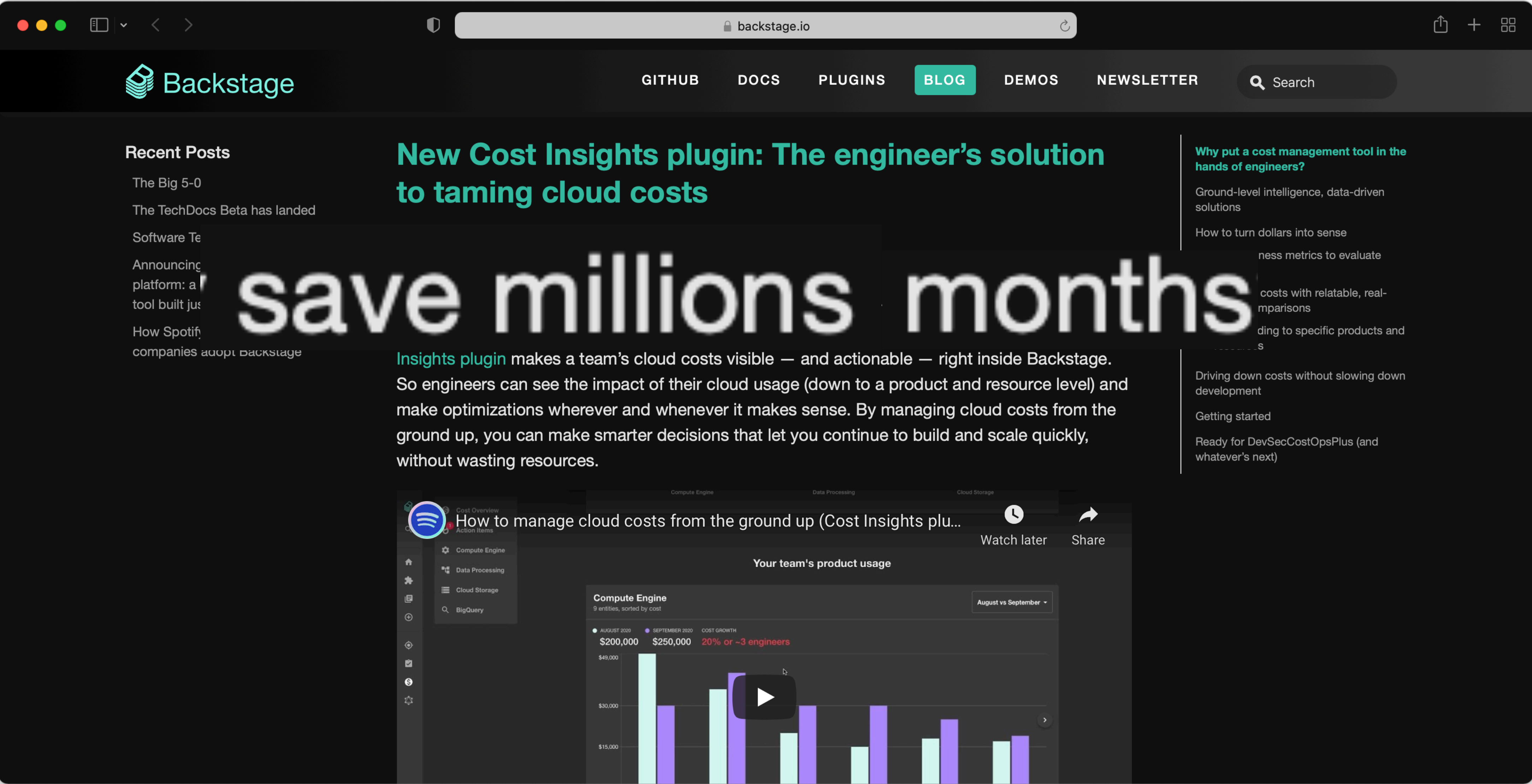




uh...  
why are our cloud  
bills so large? what **is**  
all this stuff?

mystery money pit





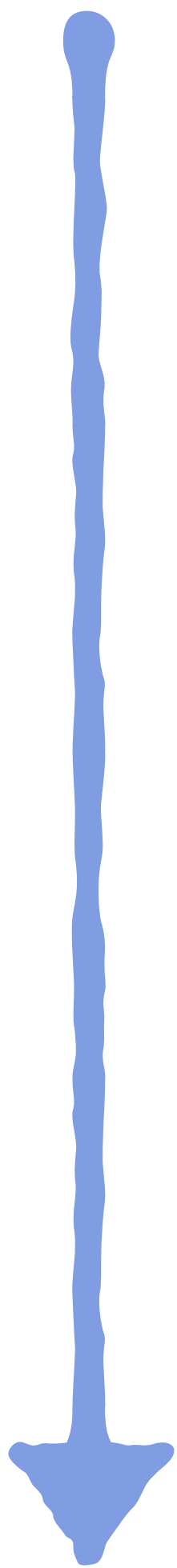
cost is now an  
engineering concern

# FinOps

getting it  
right



tangible-ness

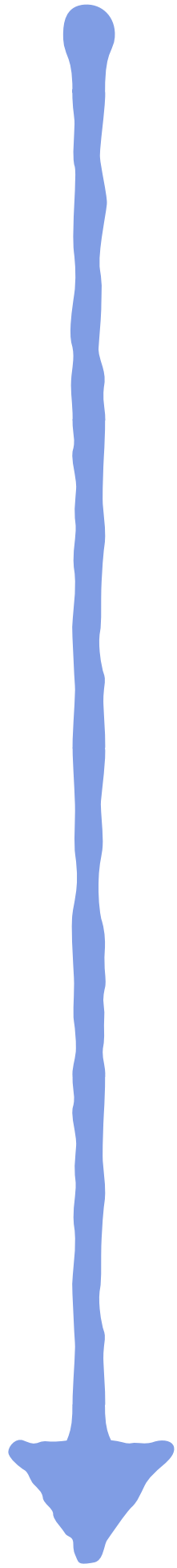


@holly\_cummins

#IBM

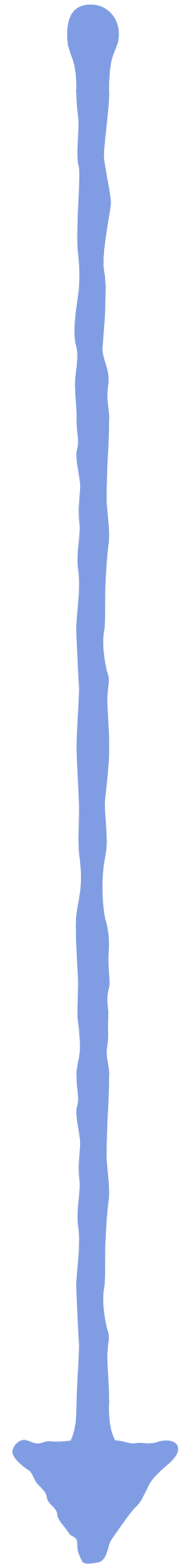
# cncf-listed products

tangible-ness



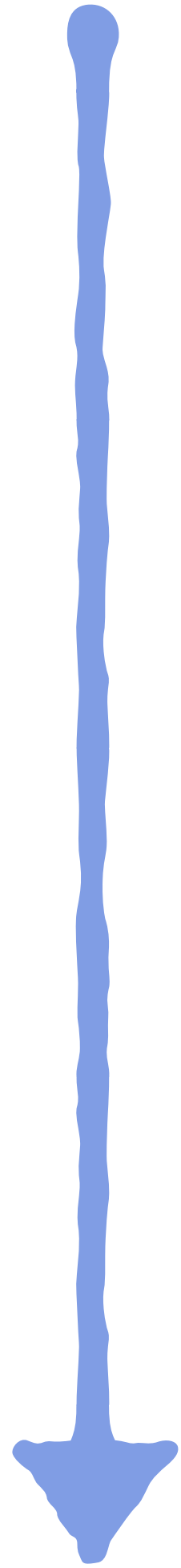
cncf-listed products  
cloud native architecture

tangible-ness

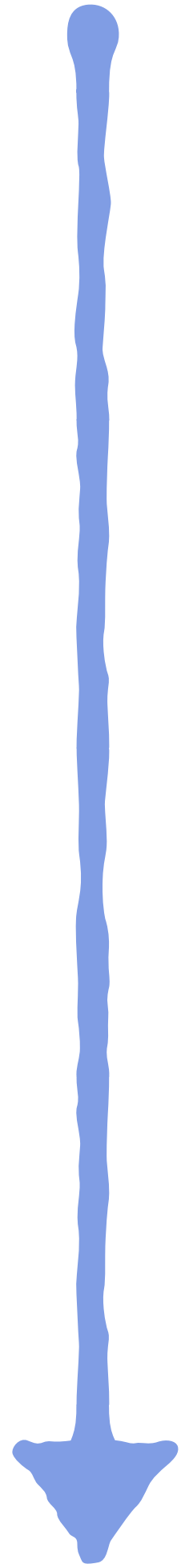


cncf-listed products  
cloud native architecture  
cloud native operations

tangible-ness

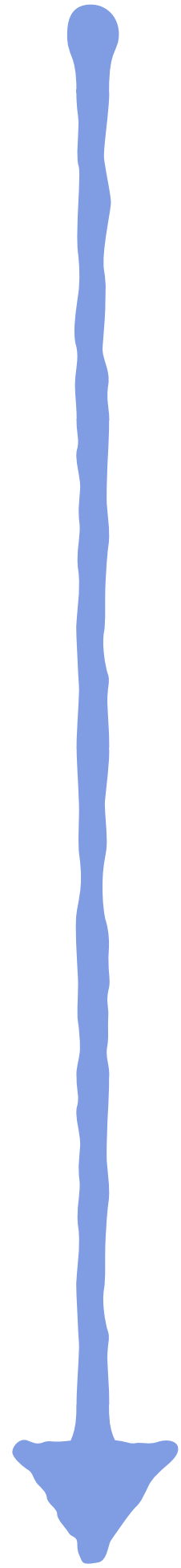


tangible-ness



cncf-listed products  
cloud native architecture  
cloud native operations  
cloud native business drivers

tangible-ness



cncf-listed products  
cloud native architecture  
cloud native operations  
cloud native business drivers

tangible-ness

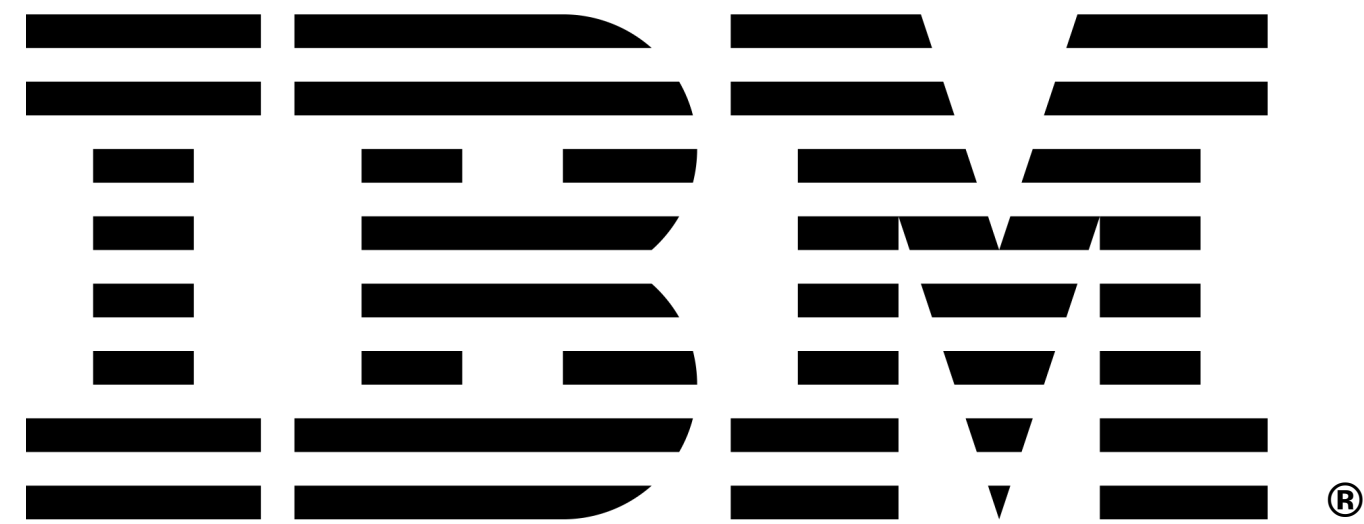
cncf-listed products

cloud native architecture

cloud native operations

cloud native business drivers

culture



@holly\_cummins