goto: copenhagen

GOTO Copenhagen 2018
Conference Nov. 19 - 21

Cloud native is about culture, **not** containers

Holly Cummins
IBM **Cloud** Garage

Follow us @gotocph

by the way ...
the IBM **Cloud** Garage is hiring ...

by the way ...
the IBM **Cloud** Garage is hiring ...

... in Copenhagen :)

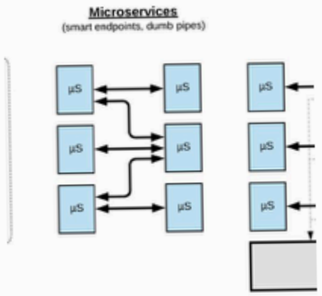
what **is** cloud native?

@holly_cummins

Daniel Bryant
@danielbryantuk

Following

I've gotta hand it to @bibryam, he's got a great way of framing things... :-)

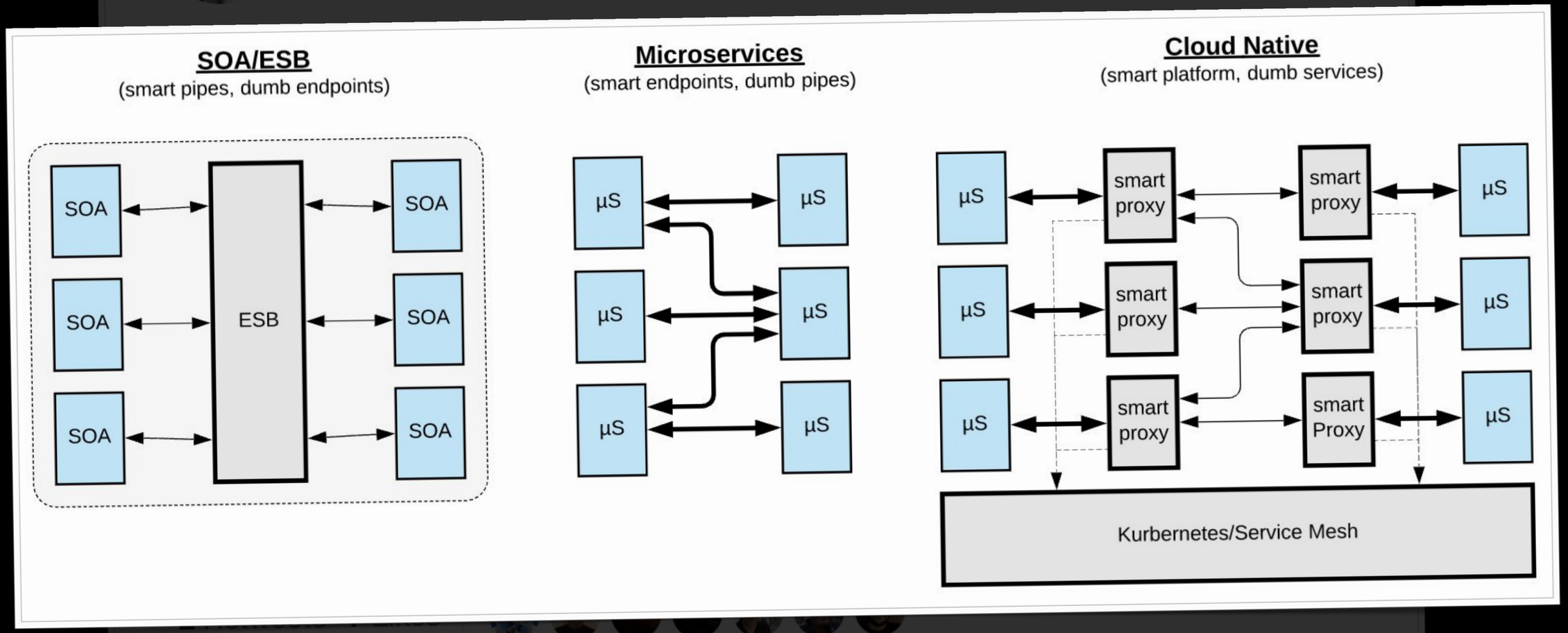


Bilgin Ibryam @bibryam
ESB -> Microservices -> CloudNative

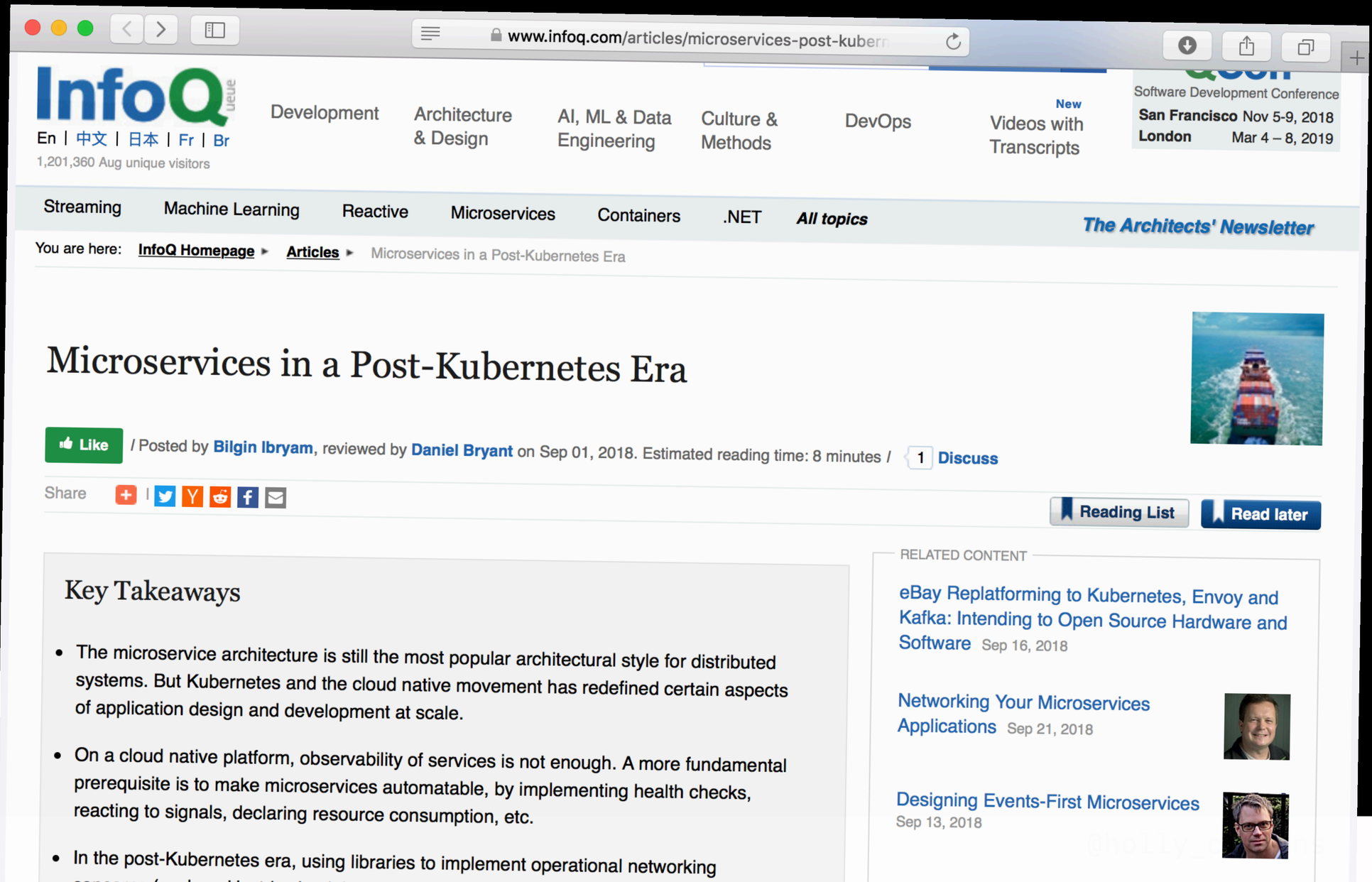
8:42 AM - 7 Aug 2018

2 Retweets **7** Likes

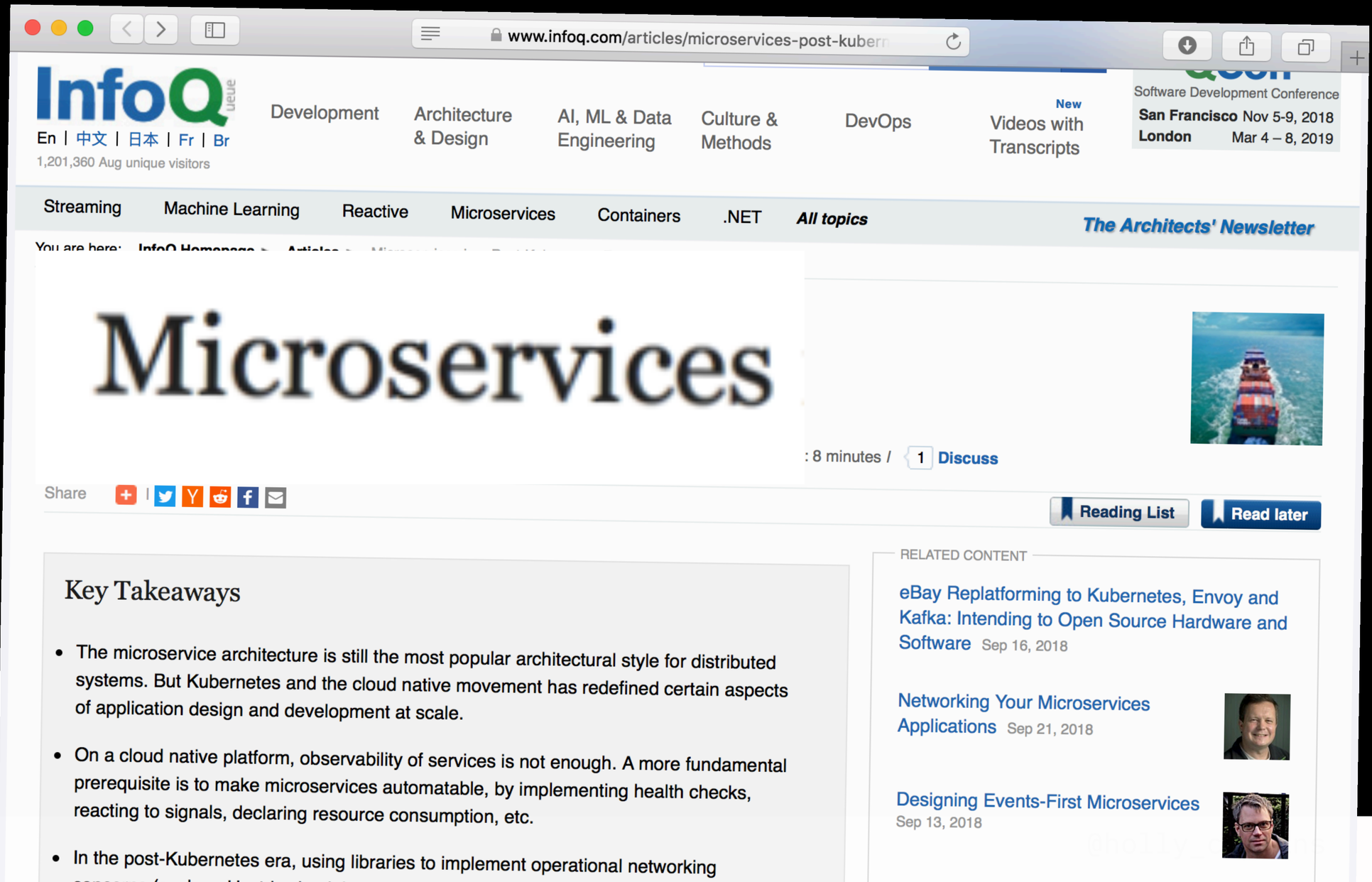
Credit to @bibryam

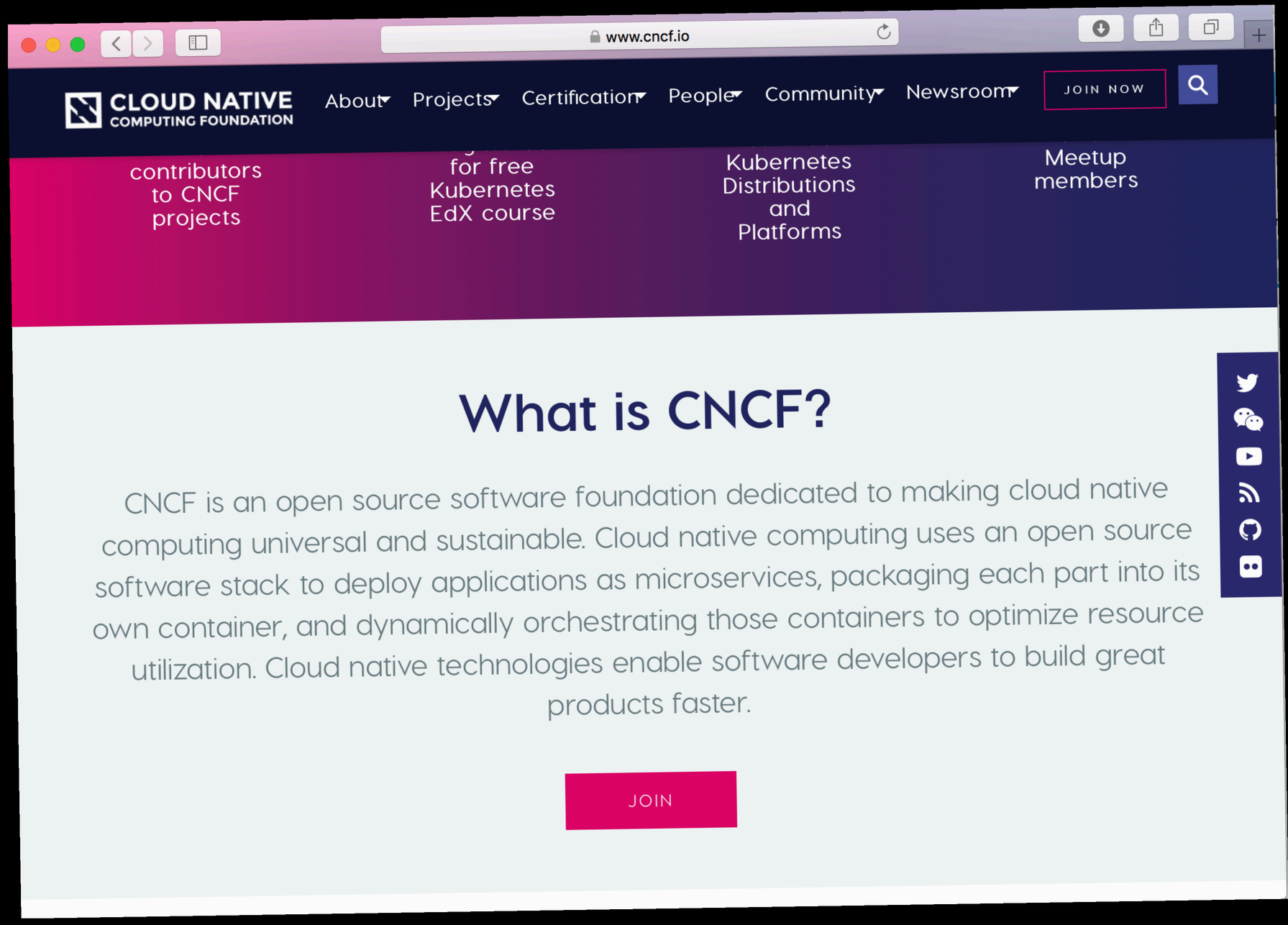


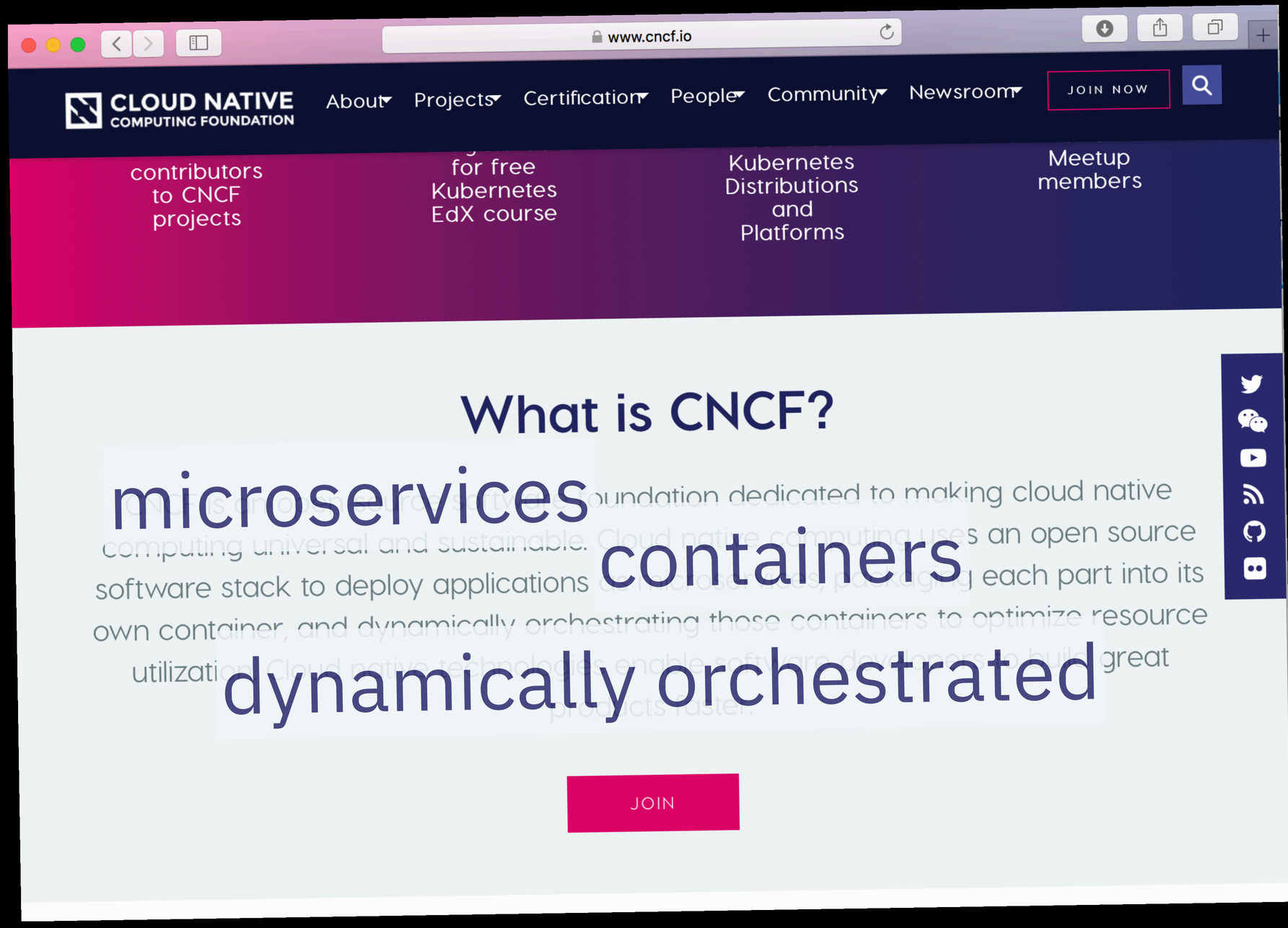
(a great article, btw)



(a great article, btw)




www.cncf.io
CLOUD NATIVE
COMPUTING FOUNDATION
About
Projects
Certification
People
Community
Newsroom
JOIN NOW
contributors
to CNCF
projects
for free
Kubernetes
EdX course
Kubernetes
Distributions
and
Platforms
Meetup
members
What is CNCF?
CNCF is an open source software foundation dedicated to making cloud native computing universal and sustainable. Cloud native computing uses an open source software stack to deploy applications as microservices, packaging each part into its own container, and dynamically orchestrating those containers to optimize resource utilization. Cloud native technologies enable software developers to build great products faster.
JOIN


www.cncf.io
CLOUD NATIVE COMPUTING FOUNDATION
About
Projects
Certification
People
Community
Newsroom
JOIN NOW
contributors to CNCF projects
for free Kubernetes EdX course
Kubernetes Distributions and Platforms
Meetup members
What is CNCF?
microservices
containers
dynamically orchestrated
JOIN

goto; copenhagen

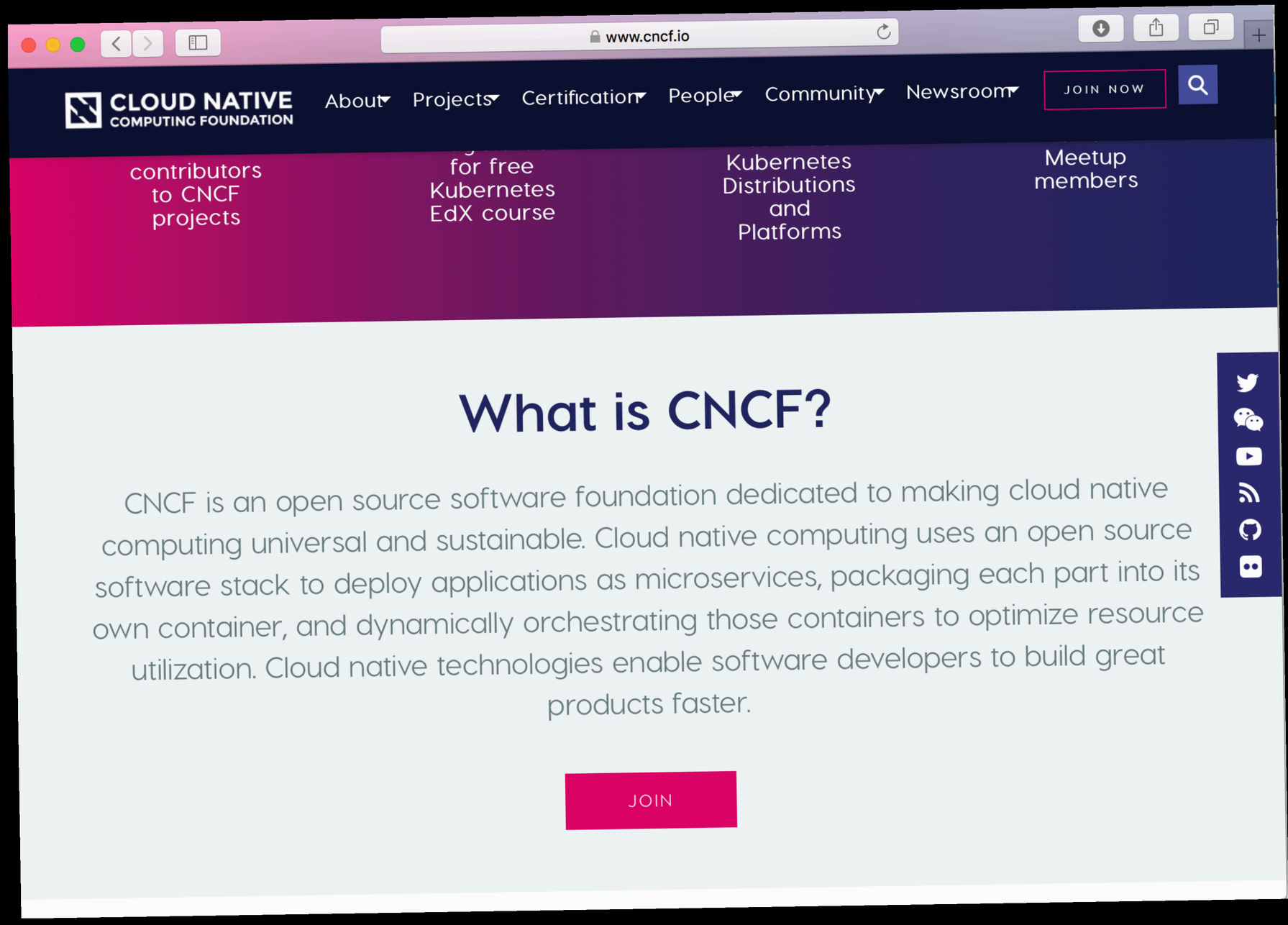
"the cloud native computing foundation is wrong ... about cloud native."

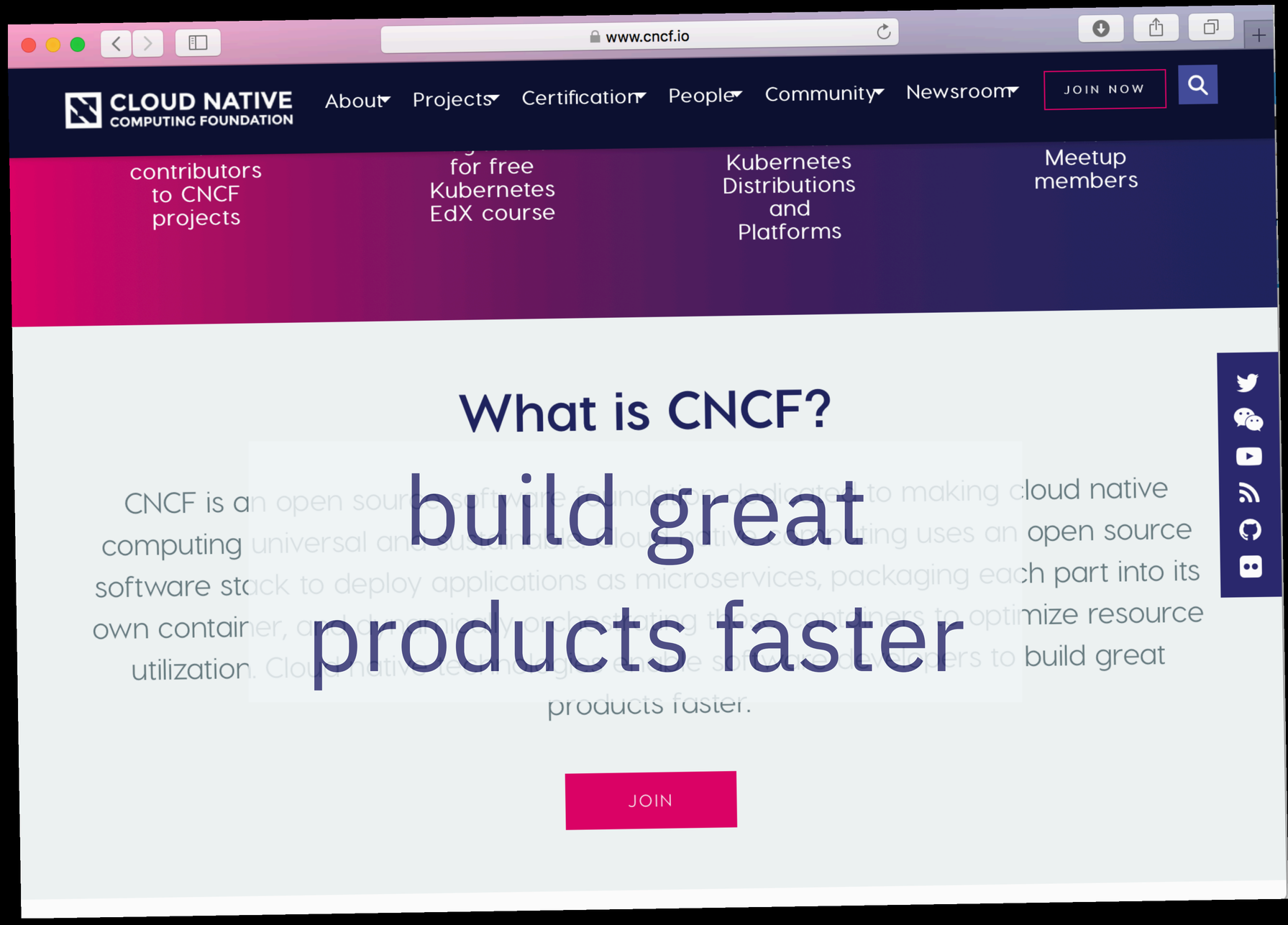
goto; copenhagen

"the cloud native computing foundation is wrong ... about cloud native."

Holly

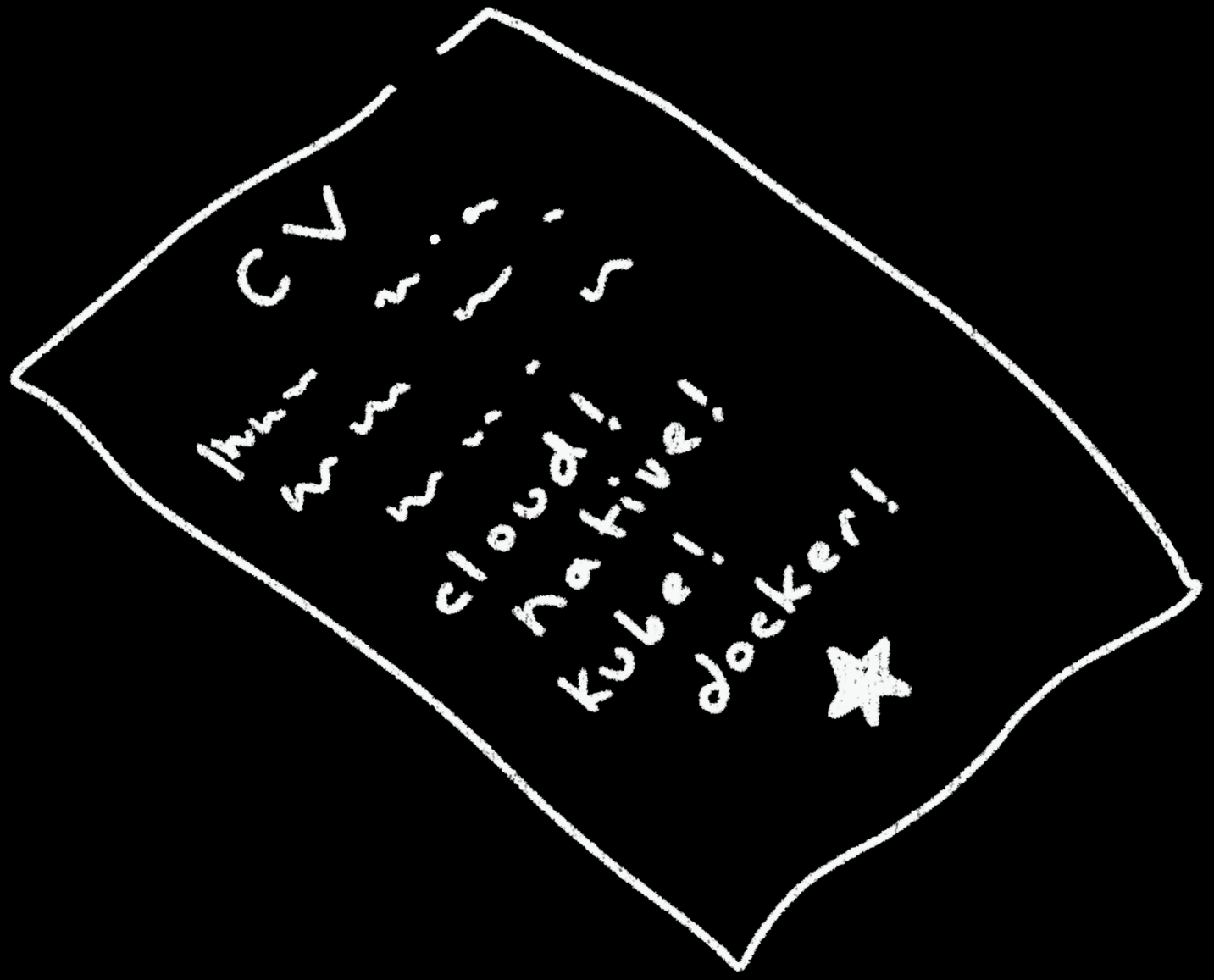
Kasper Nissen @GOTOCPH
@phennex
Follow
If you are attending #gotocph and want to discuss @CloudNativeFdn tech, please feel free to PM me or find me at the venue the next couple of days. #cloudnative
2:34 PM - 18 Nov 2018 from Copenhagen, Denmark
4 Retweets 4 Likes
4
4


www.cncf.io
CLOUD NATIVE
COMPUTING FOUNDATION
About
Projects
Certification
People
Community
Newsroom
JOIN NOW
contributors
to CNCF
projects
for free
Kubernetes
EdX course
Kubernetes
Distributions
and
Platforms
Meetup
members
What is CNCF?
CNCF is an open source software foundation dedicated to making cloud native computing universal and sustainable. Cloud native computing uses an open source software stack to deploy applications as microservices, packaging each part into its own container, and dynamically orchestrating those containers to optimize resource utilization. Cloud native technologies enable software developers to build great products faster.
JOIN


www.cncf.io
CLOUD NATIVE
COMPUTING FOUNDATION
About
Projects
Certification
People
Community
Newsroom
JOIN NOW
contributors to CNCF projects
for free Kubernetes EdX course
Kubernetes Distributions and Platforms
Meetup members
What is CNCF?
build great products faster
JOIN

why?

what **problem** are
we trying to solve?


CV
cloud!
native!
kube!
docker!

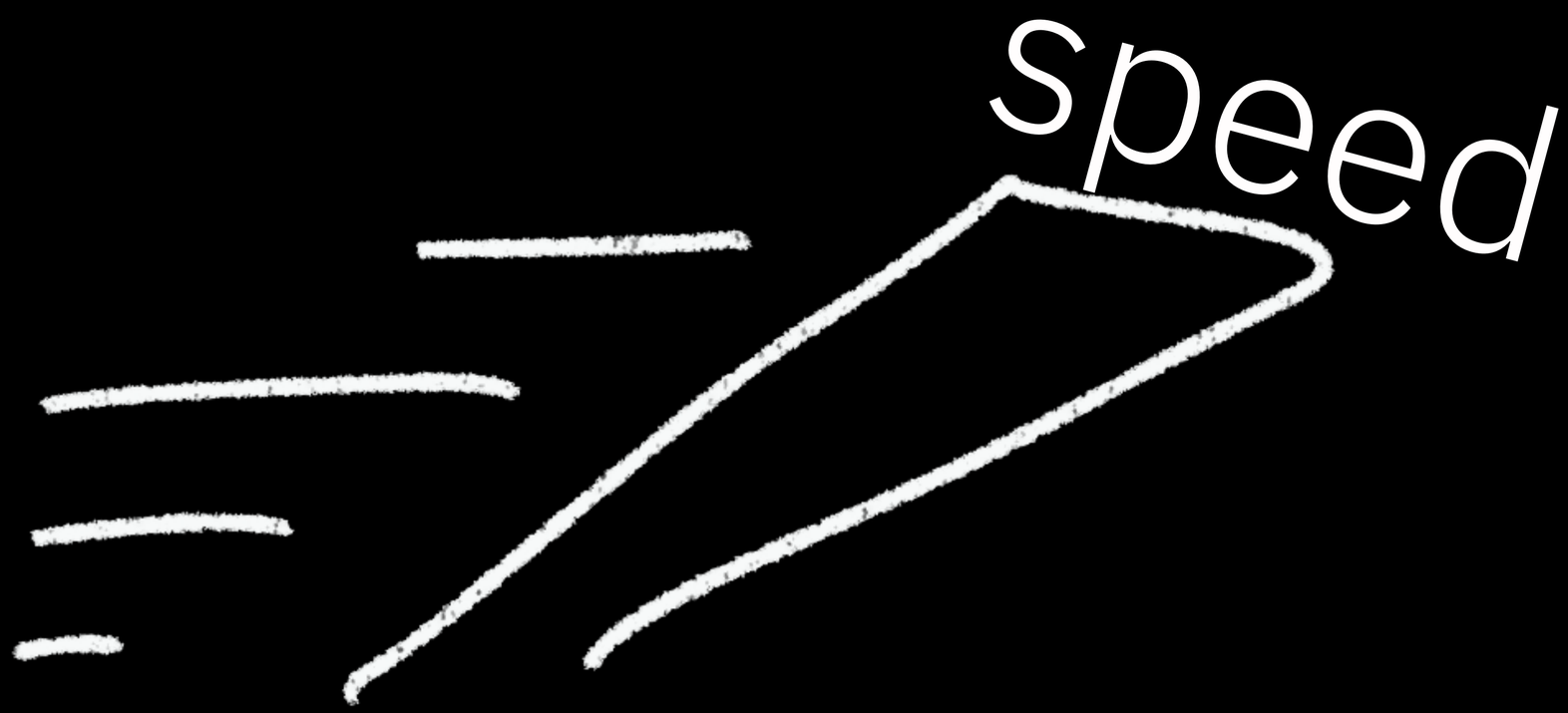
CV-driven development.

“my CV looks dull”
is not a good reason
to go cloud native

why cloud?

cost

e l a s t i c i t y


speed

exotic capabilities

why cloud native?

12

factors

12 factors

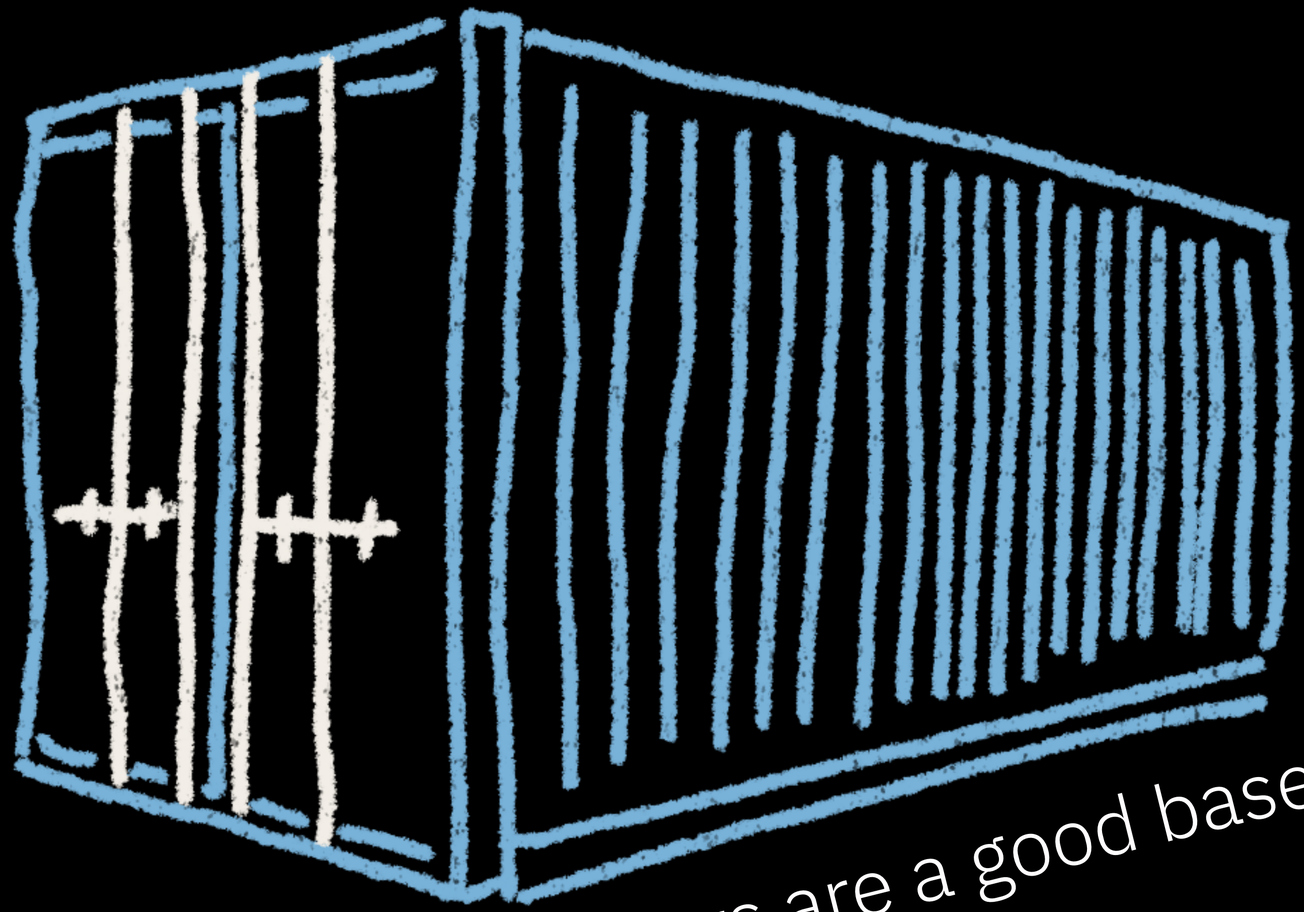
how to write a cloud application so you don’t get electrocuted

cloud native is not a synonym for ‘microservices’

if ‘cloud native’ has to be a synonym for anything, it would be ‘idempotent’

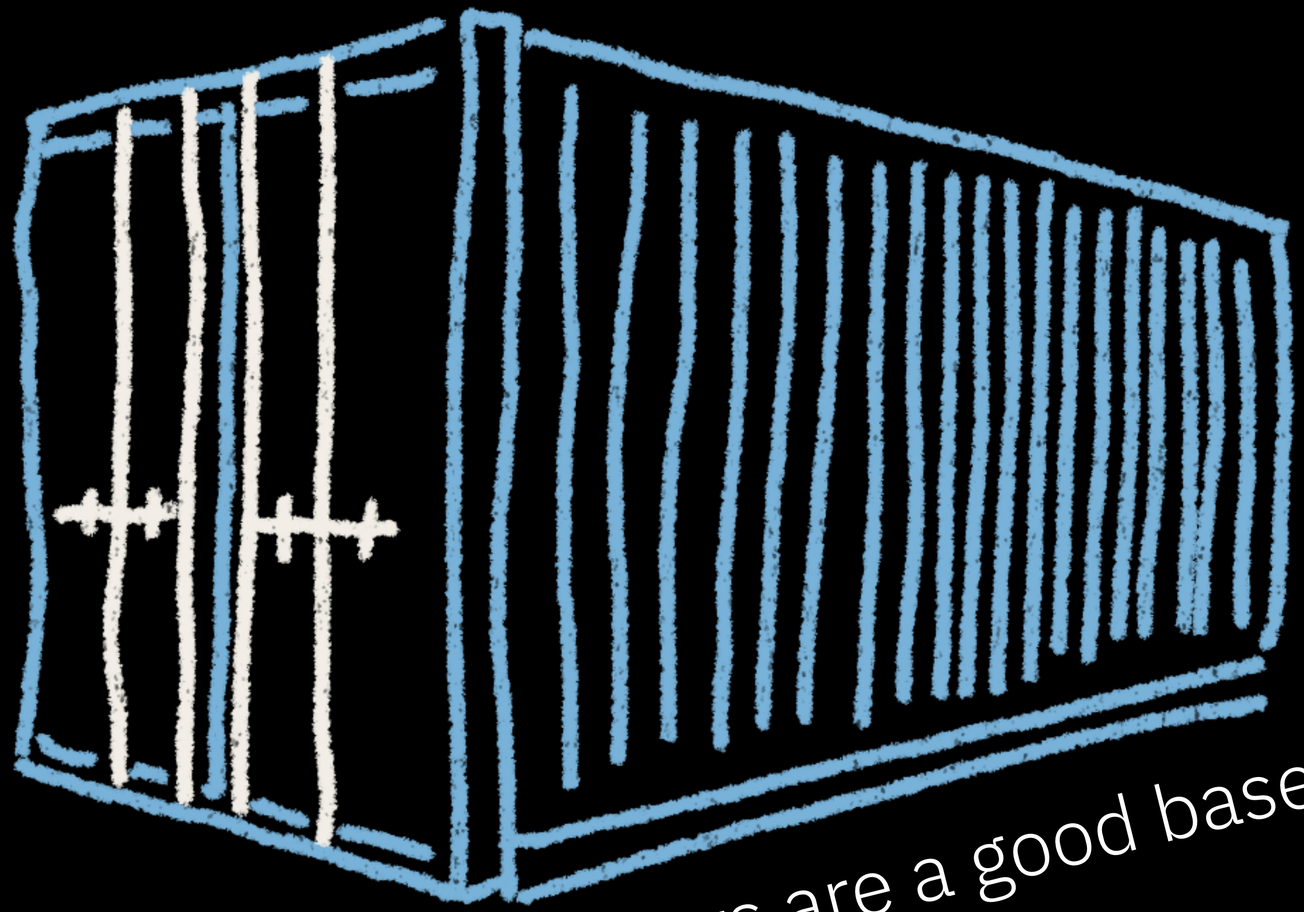
if ‘cloud native’ has to be a synonym for anything, it would be ‘idempotent’

which definitely needs a synonym



containers are a good base

it’s not a **competition** to see how many you can have



containers are a good base

you do not need intra-app http communication to be cloud native

complexity adds expense

unnecessary complexity
adds unnecessary expense

space pencil

space pencil

$128.89

space pencil

$128.89

space pen

space pencil

$128.89

space pen

$2.39

space pencil

$128.89

+medical bills

space pen

$2.39

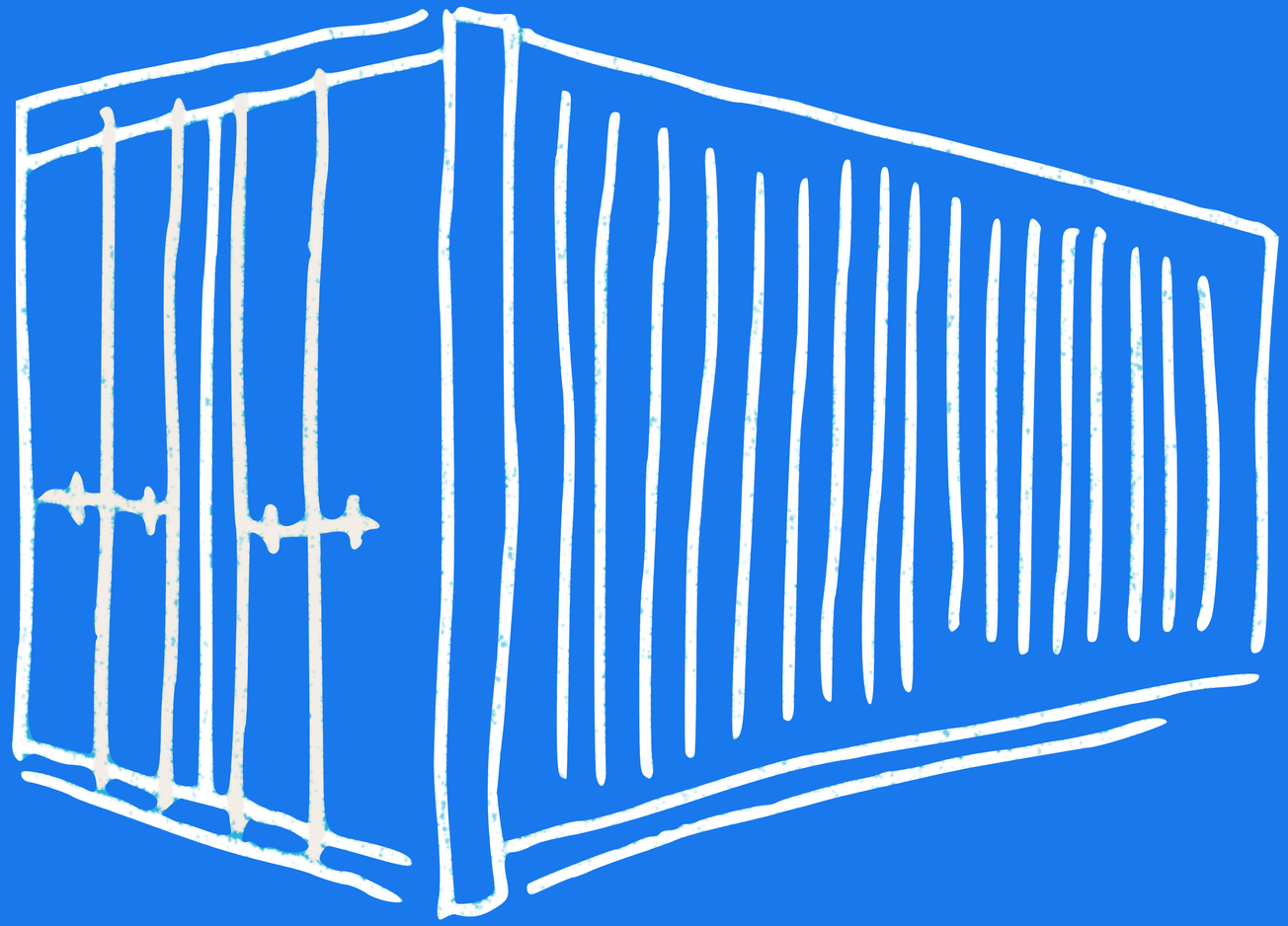
accidental complexity

essential complexity

accidental complexity

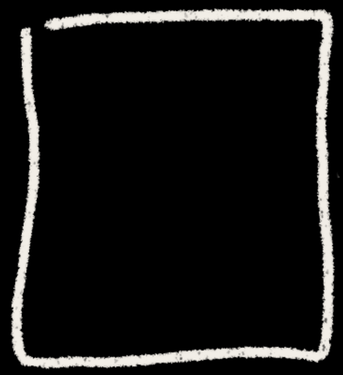
essential complexity

“this will impress my boss” complexity



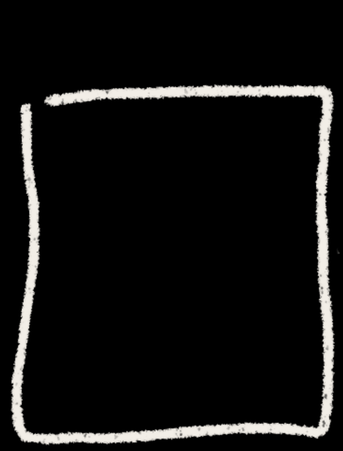
what **is** a container?

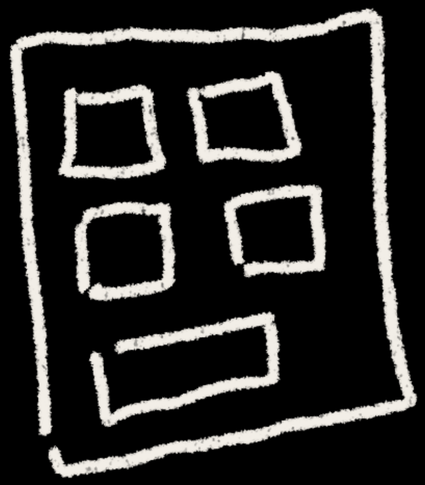
bare metal



bare metal

virtual machines

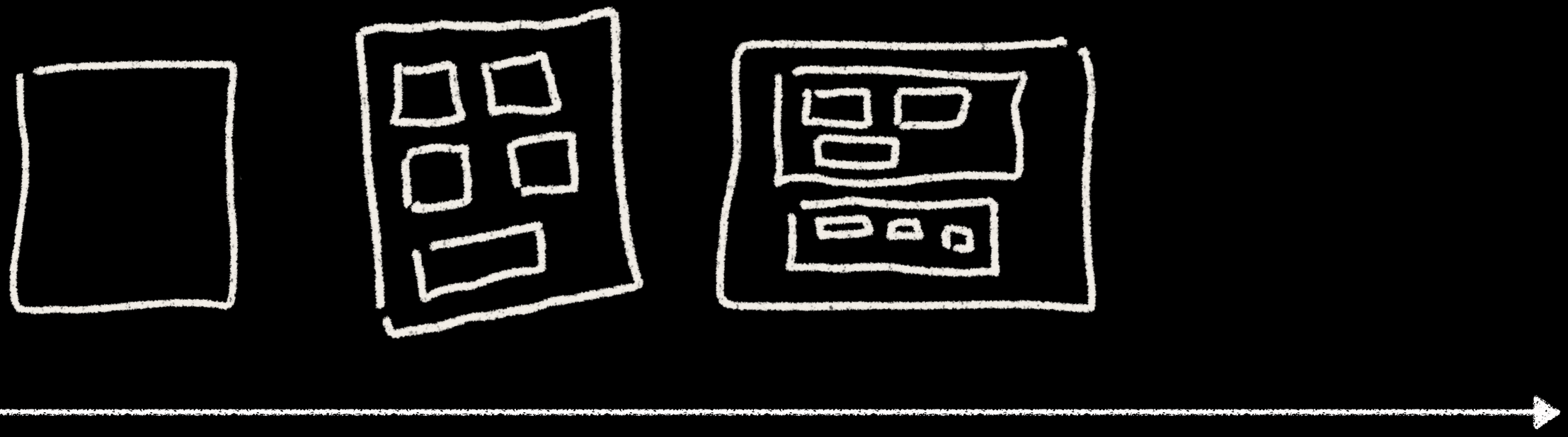




bare metal

virtual machines

containers

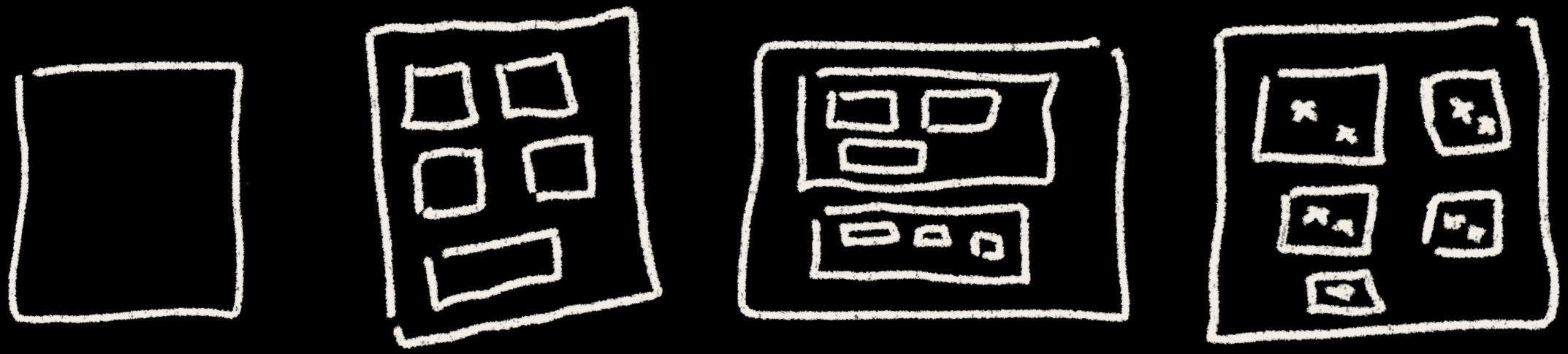


bare metal

virtual machines

containers

functions



funtainers

function-tainers

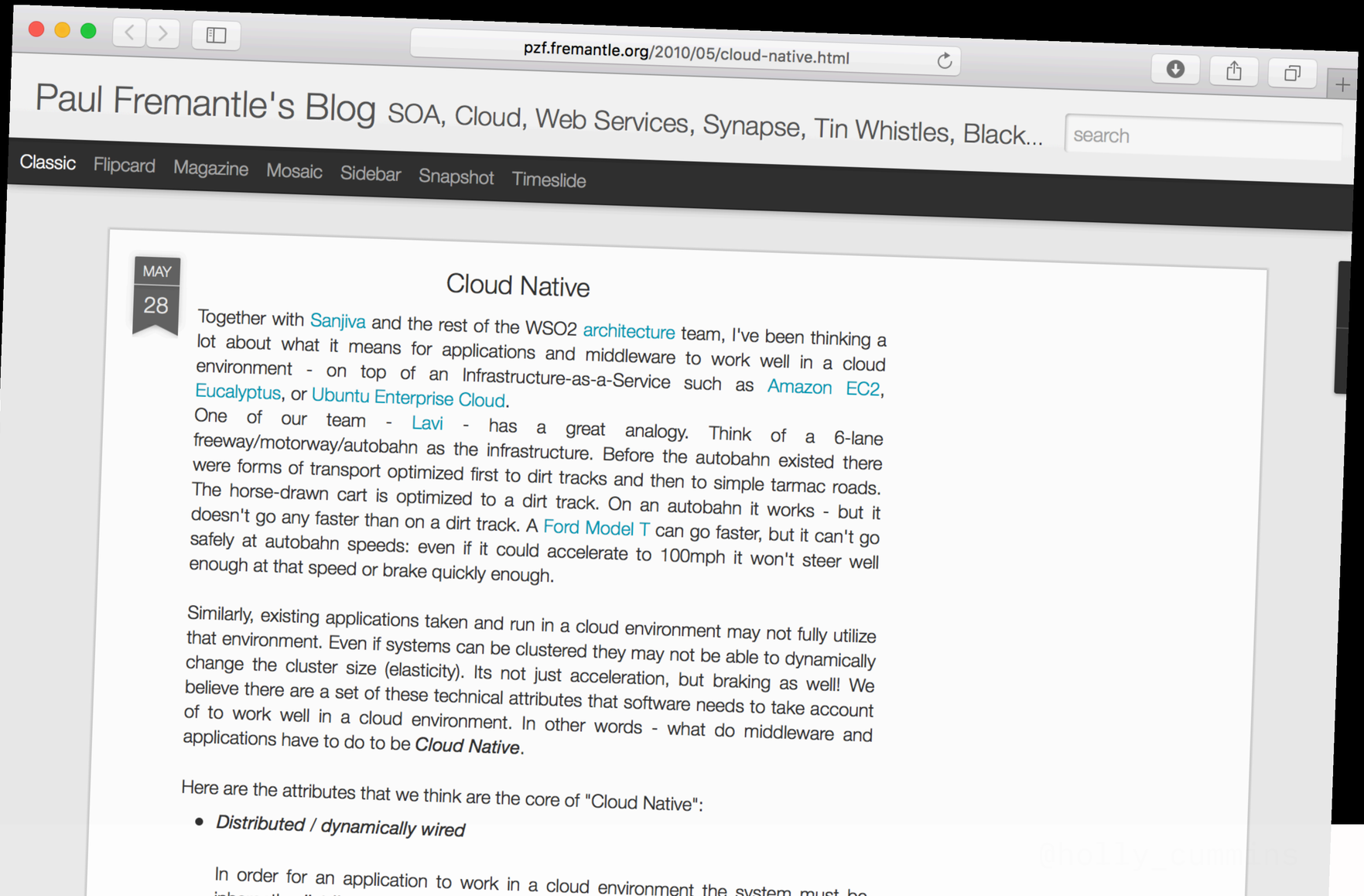
funtainers

Thermos® FUNtainer®

funtainers

all the fun of containers, but without having to worry about kubernetes

2010
the dawn of
cloud native



#IBMCloudGarage

behaves well on the cloud

behaves well on the cloud

written for the cloud

this is all how we **run** our application, not what's **in** it

speed

speed

what’s the point of getting the same old **stuff** to market faster?

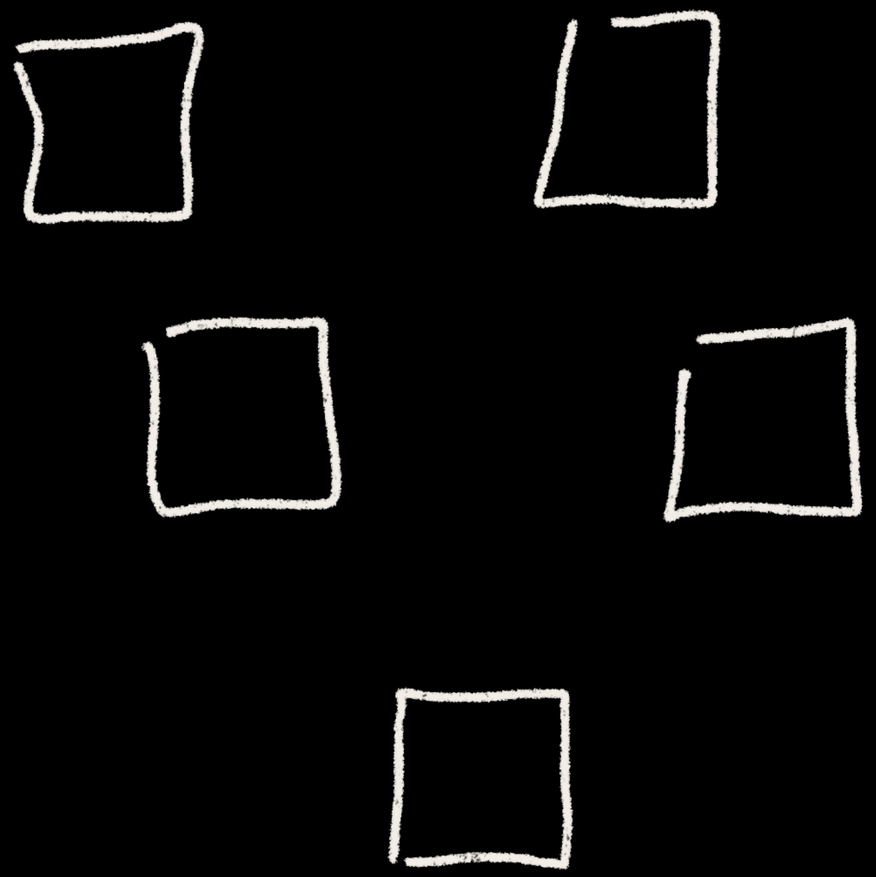
what’s the point of being able to respond to the market, if you **don’t?**

what’s the point of architecture that can go faster, if you don’t go faster?

how to fail at cloud native

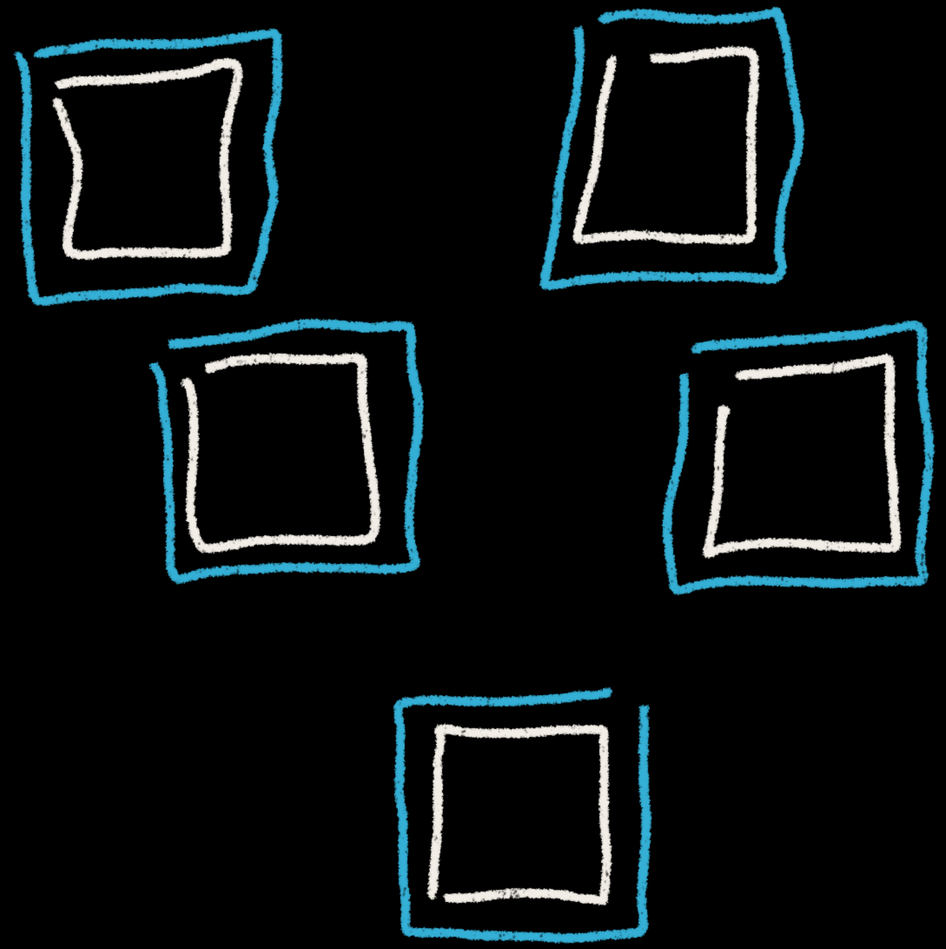
“each of our microservices has duplicated the same object model … with twenty classes and seventy fields”

Microservice



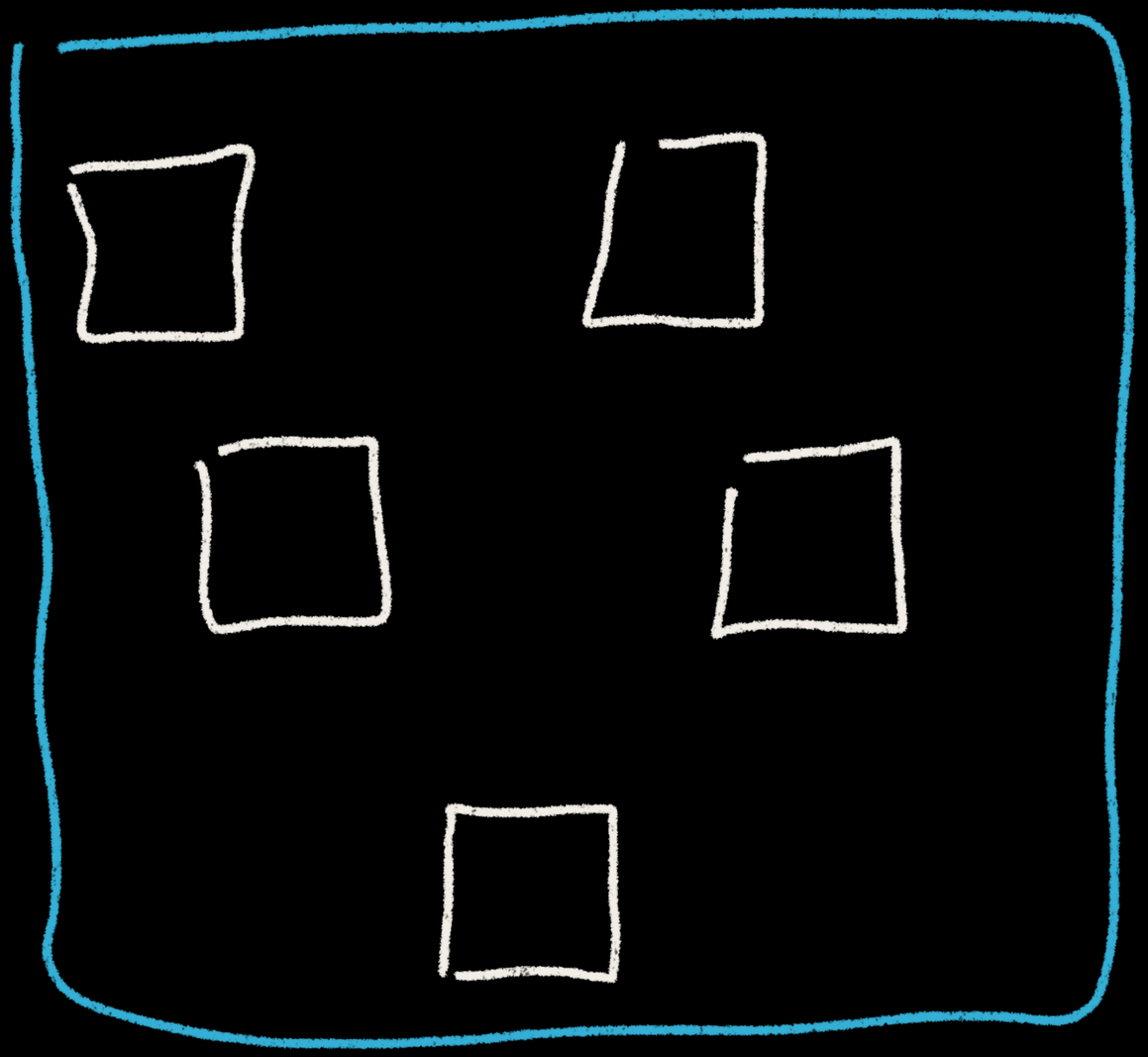
Microservice

Domain



Microservice

Domain



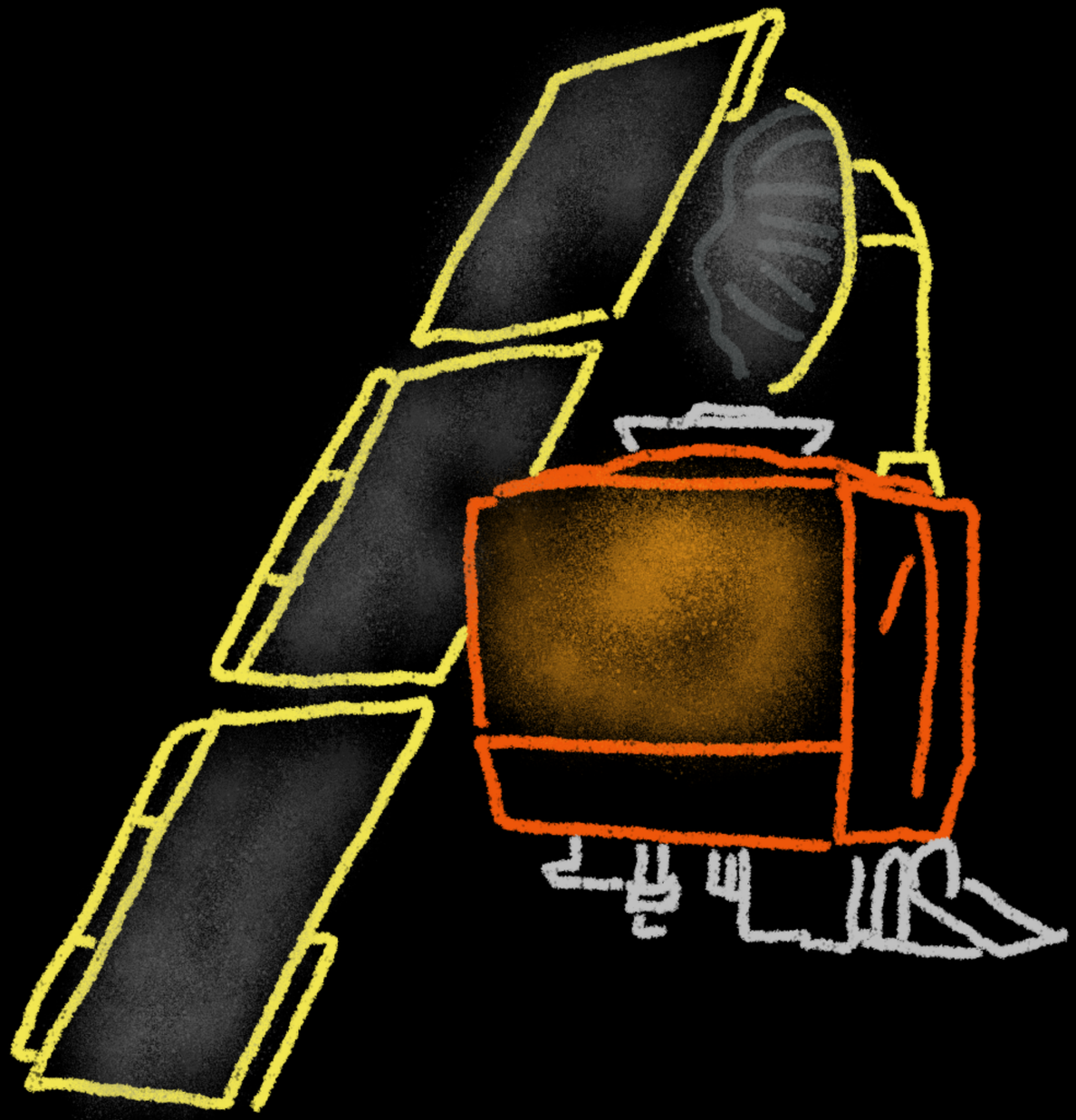
“every time we change code, something breaks”

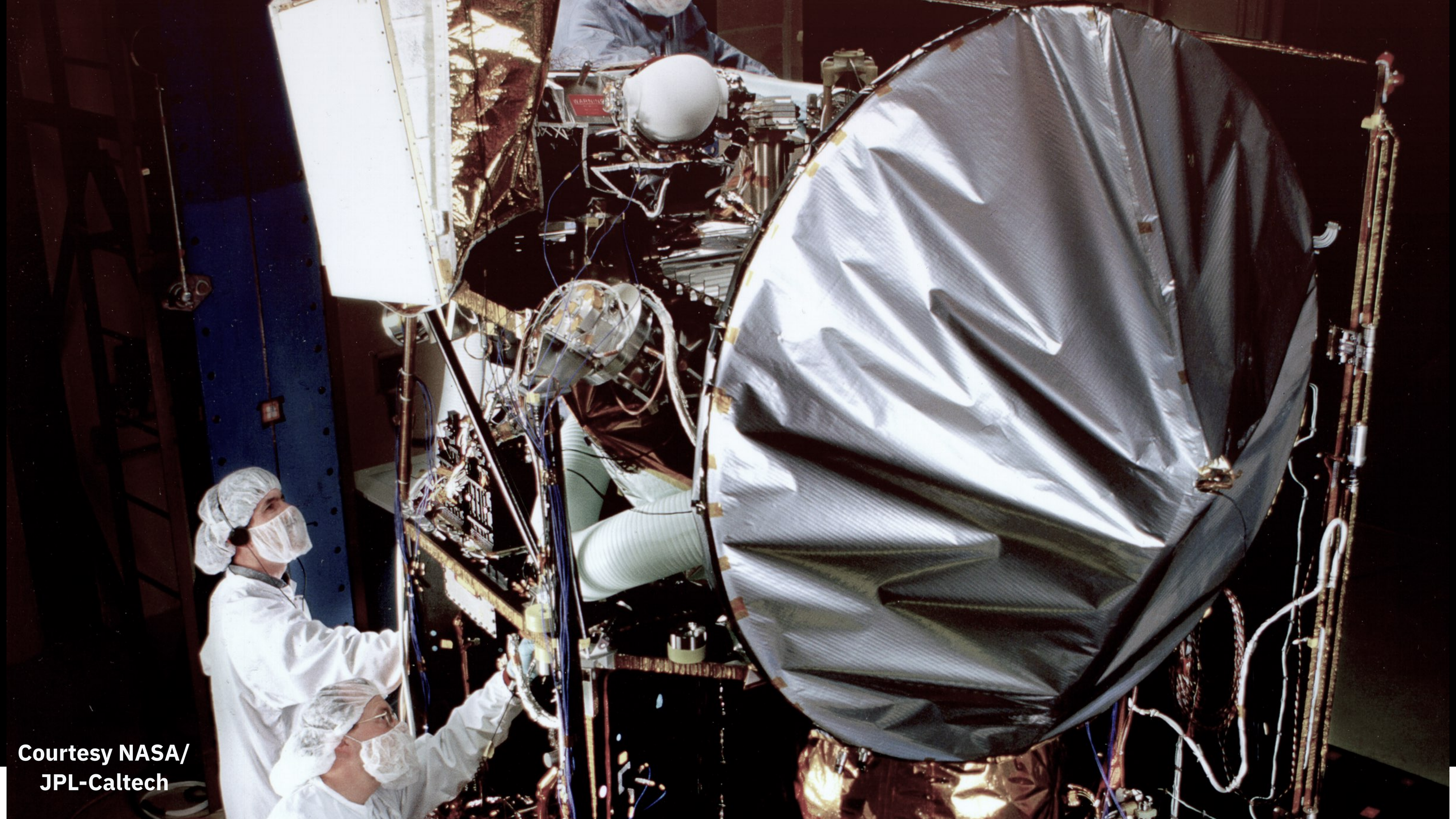
distributed monolith

cloud-native spaghetti is still spaghetti

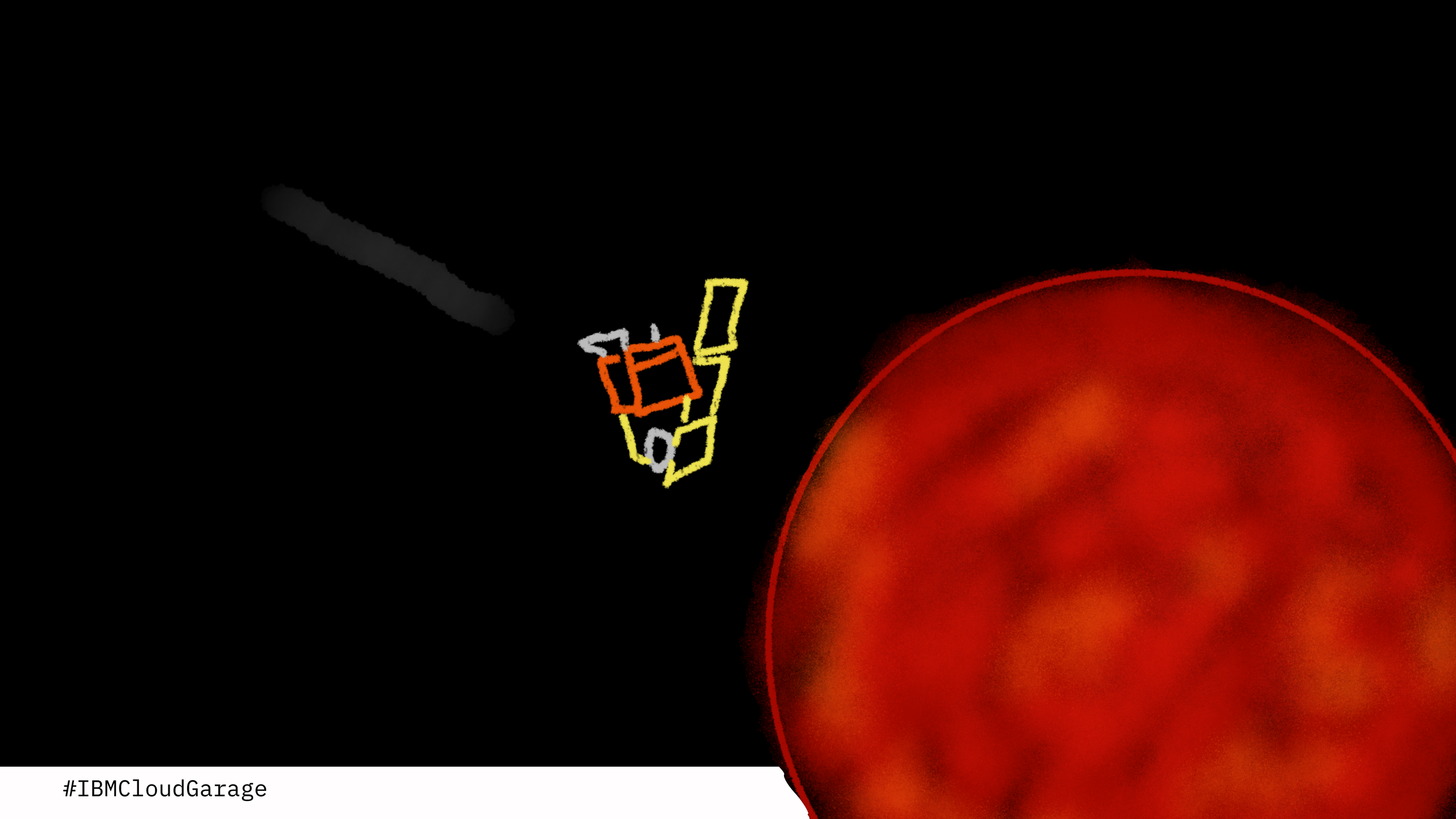
(Image: Cloudy with a Chance of Meatballs.)

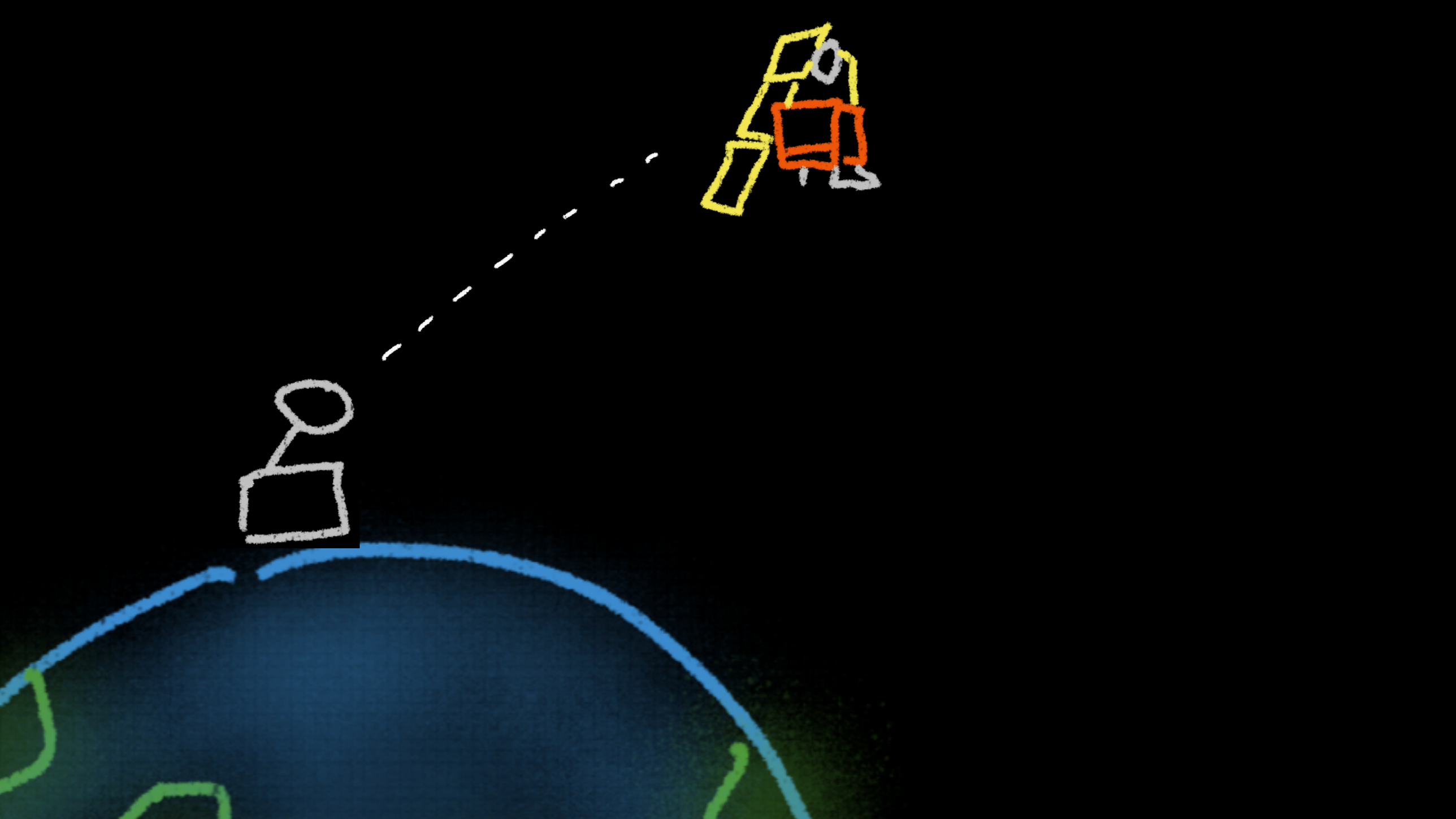
just because a system runs across 6 containers doesn’t mean it’s decoupled

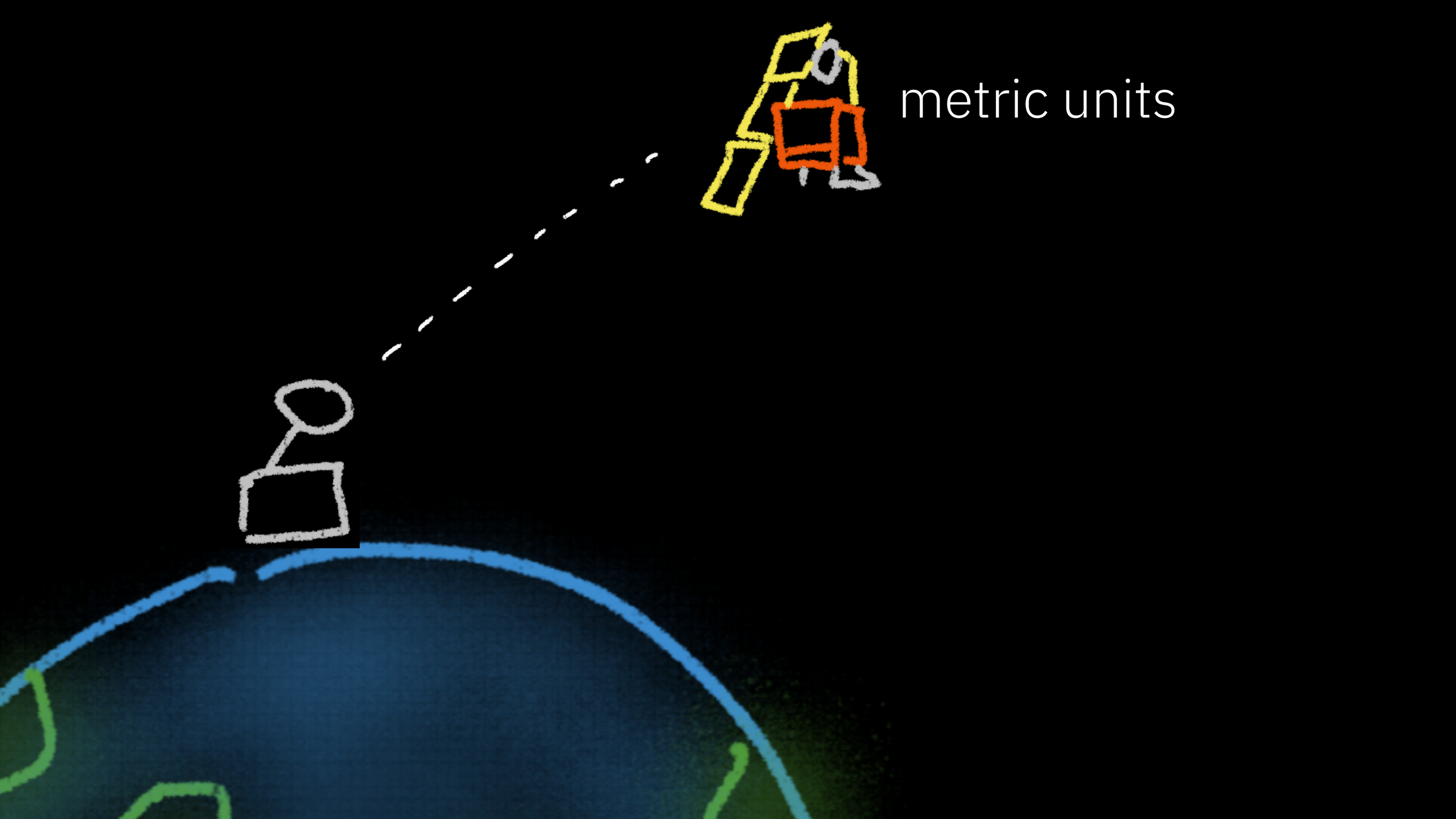


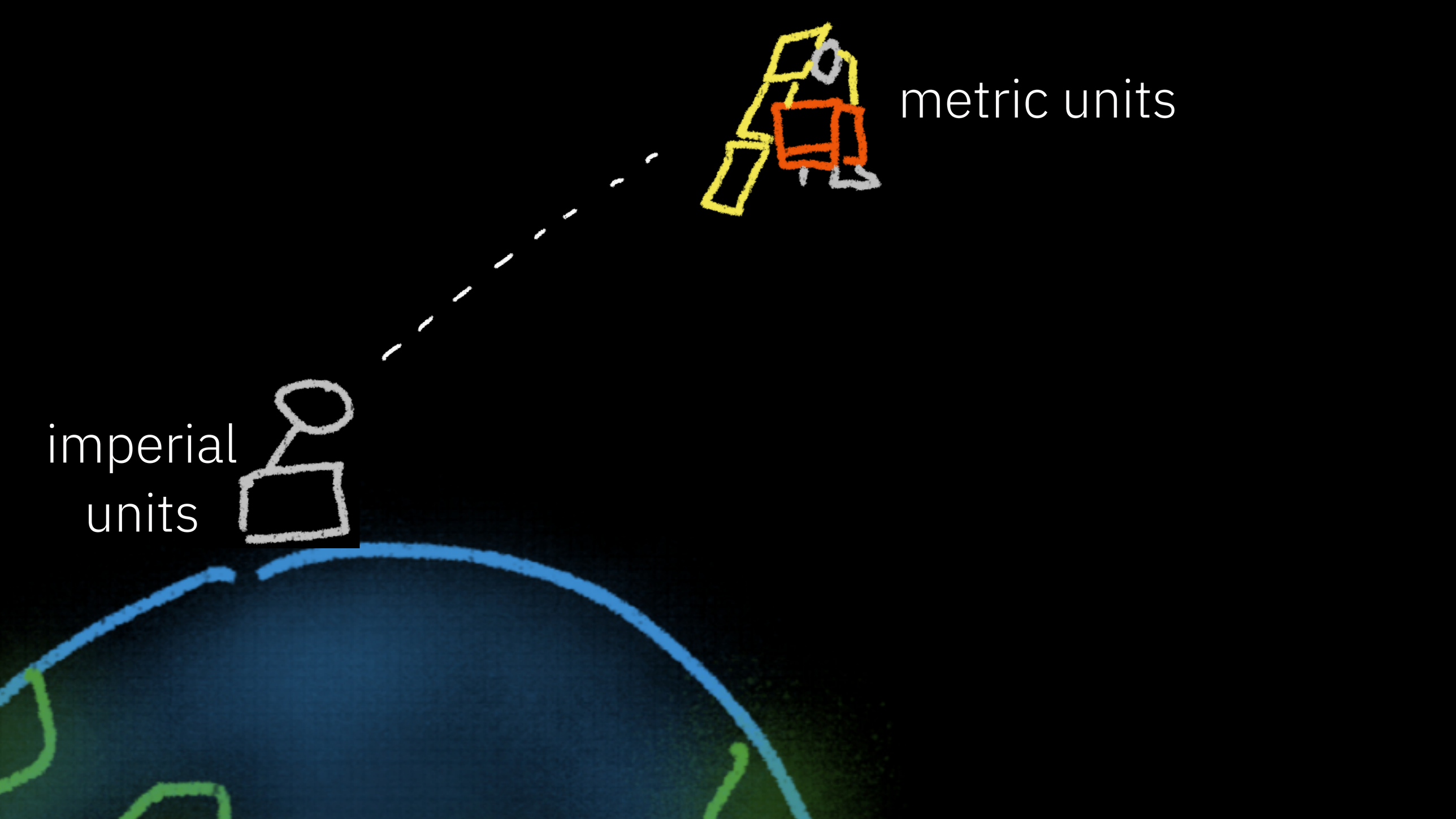


Courtesy NASA/
JPL-Caltech






metric units


metric units
imperial
units

metric units
imperial units
distributing did not help

microservices **need**
consumer-driven contract tests

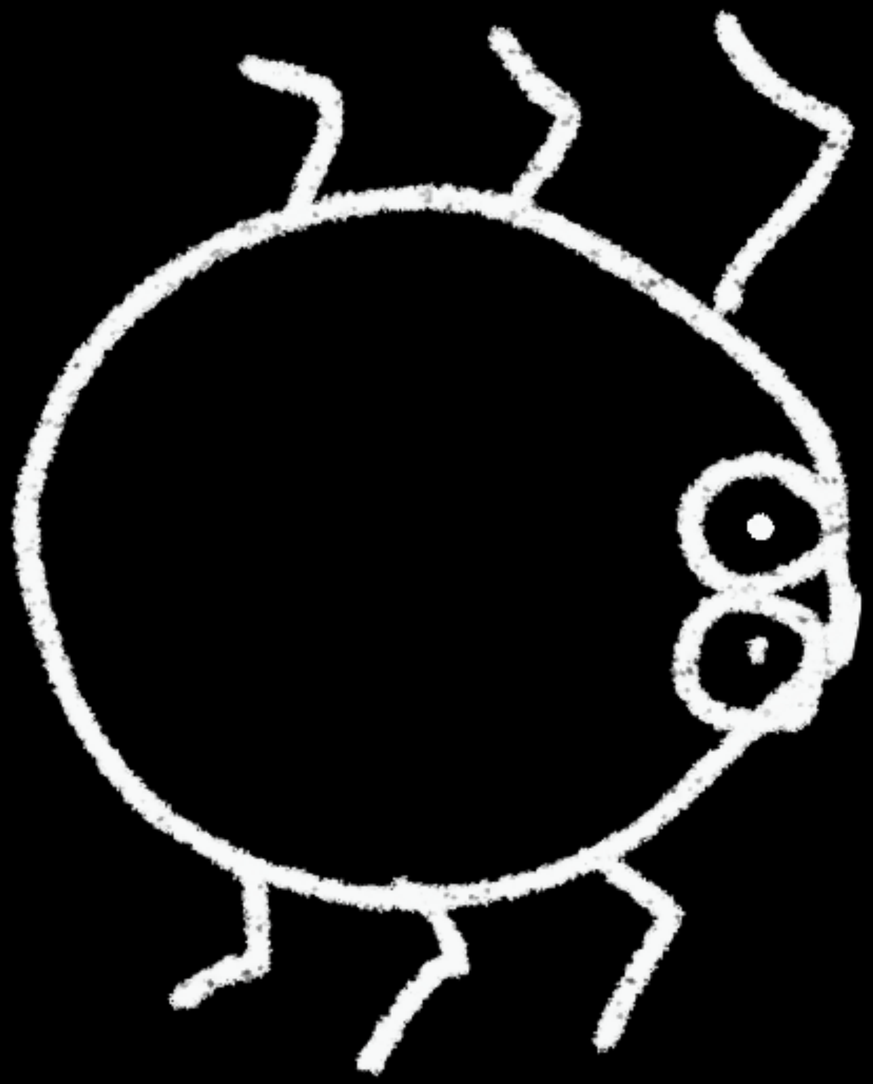
“our tests aren’t automated”

“we don’t know if
our code works”

systems **will** behave in unexpected ways

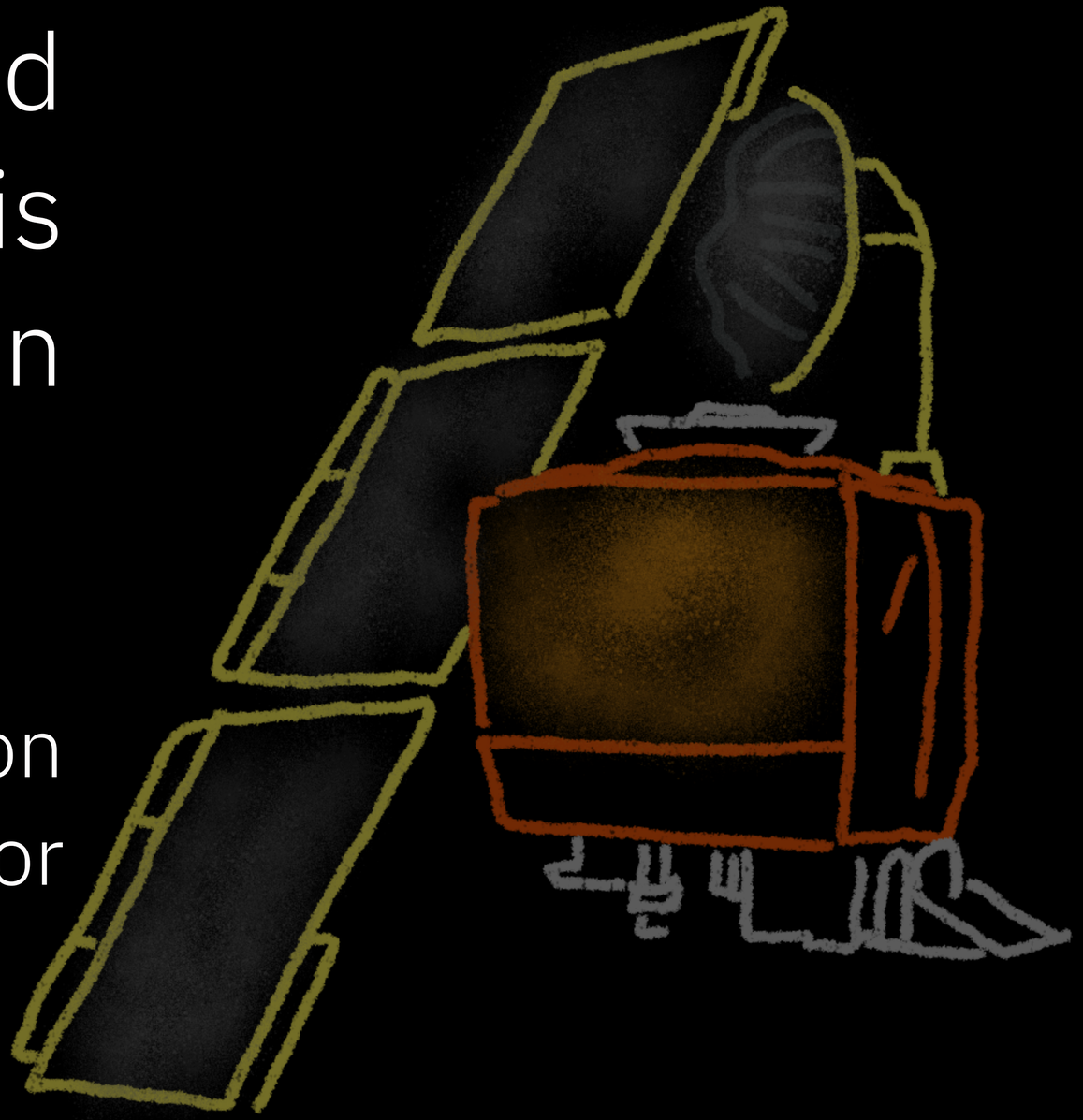
documentation can be wrong

dependency updates
can change behaviour

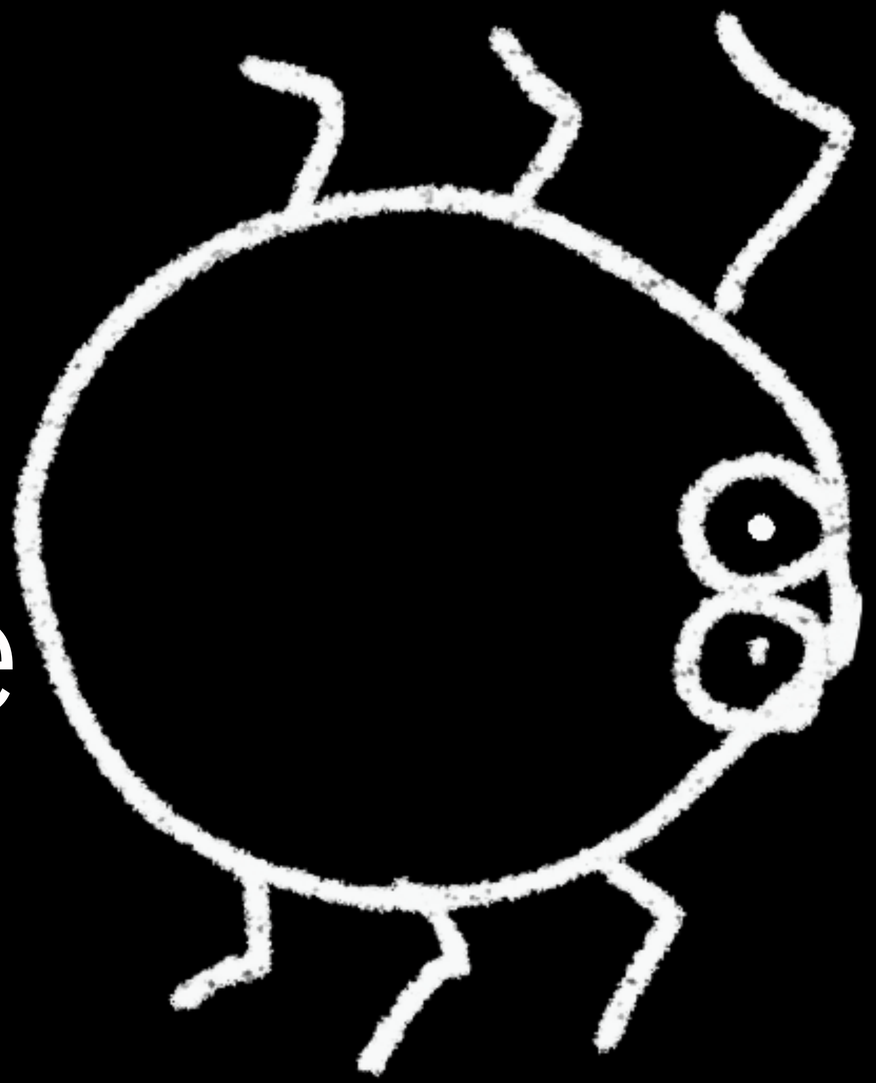


“Had we done end-to-end testing, we believe this error would have been caught.”

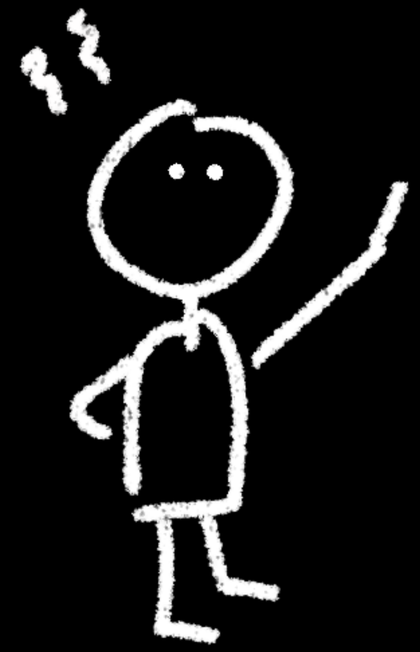
Arthur Stephenson
Chief Investigator



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



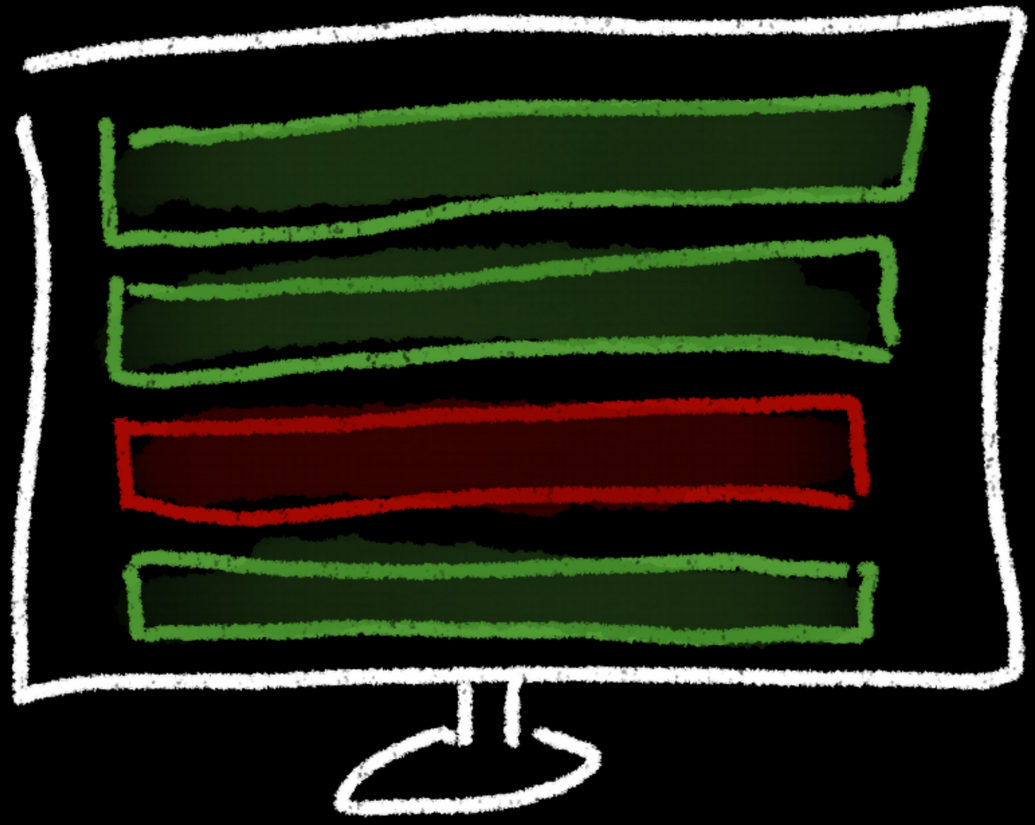
microservices **need**
automated integration tests



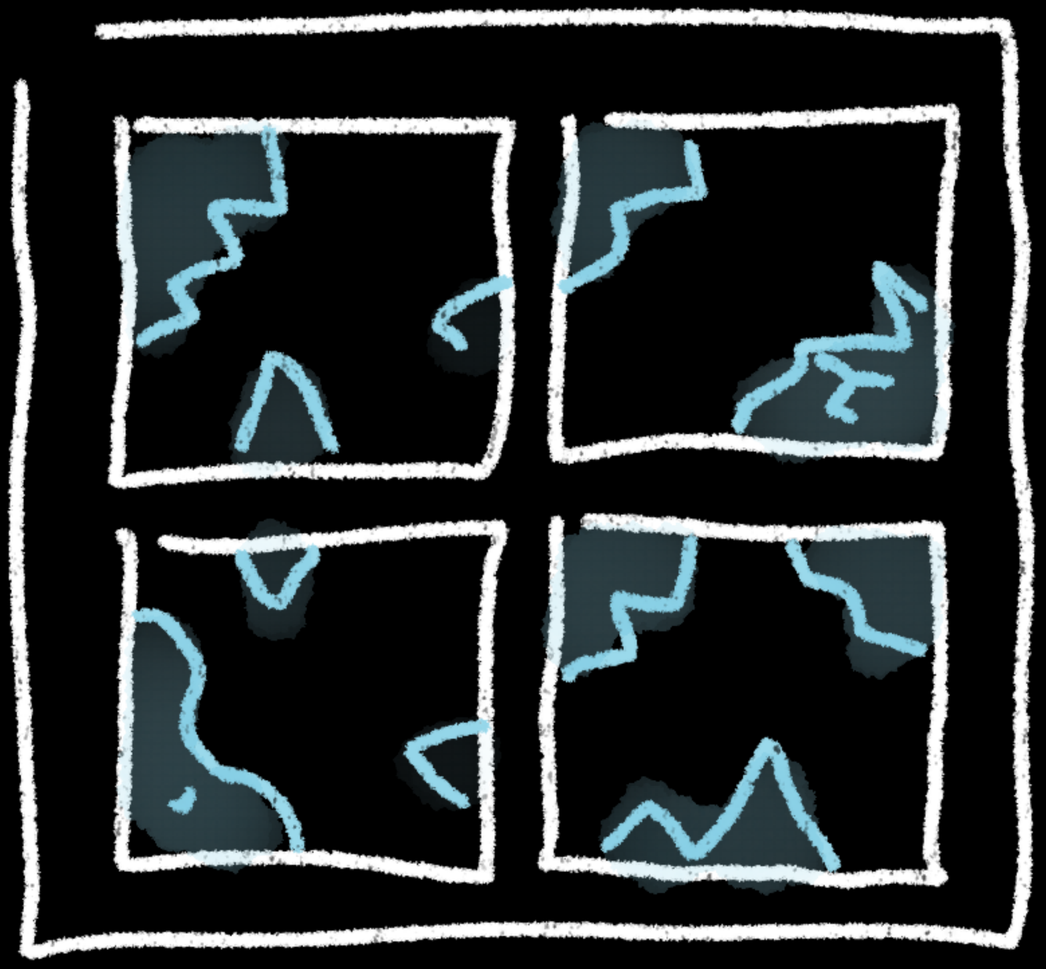
not a good CI/CD indicator

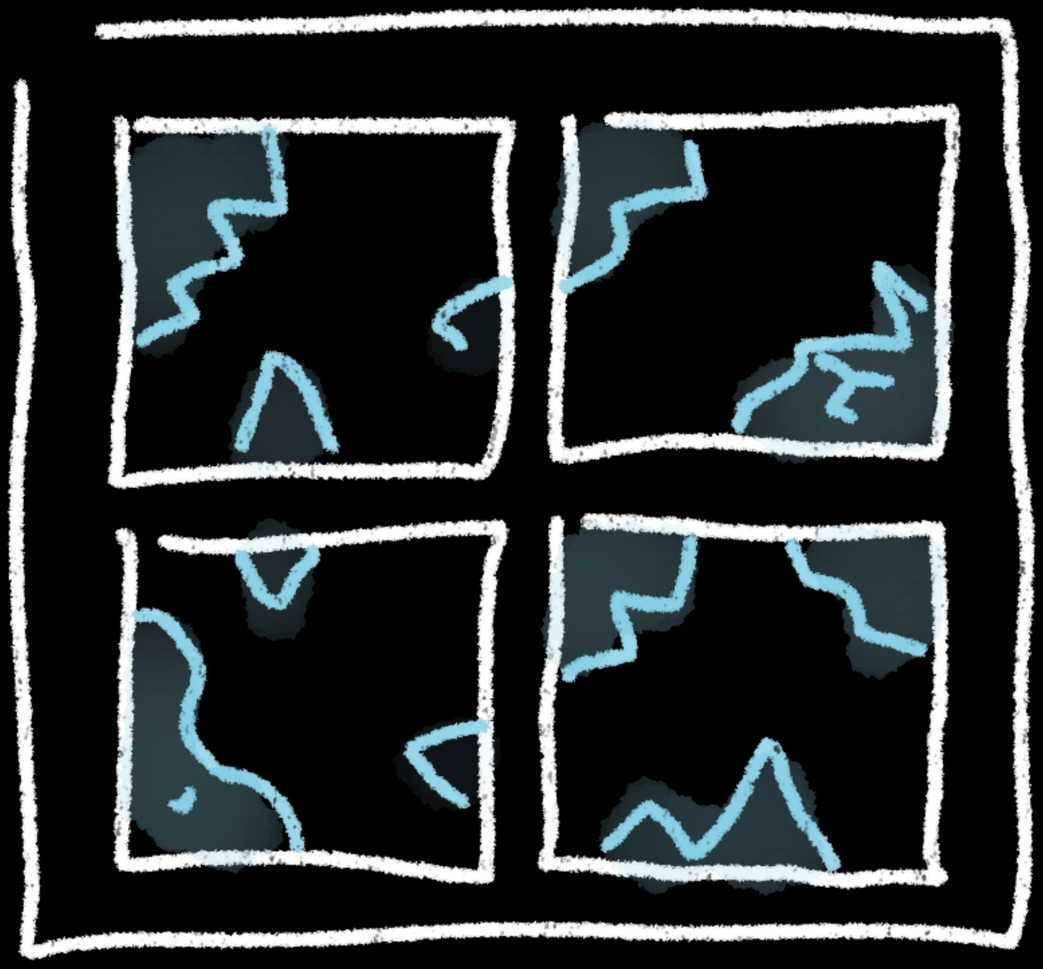
a good CI/CD indicator

“we don’t know when the build is broken”



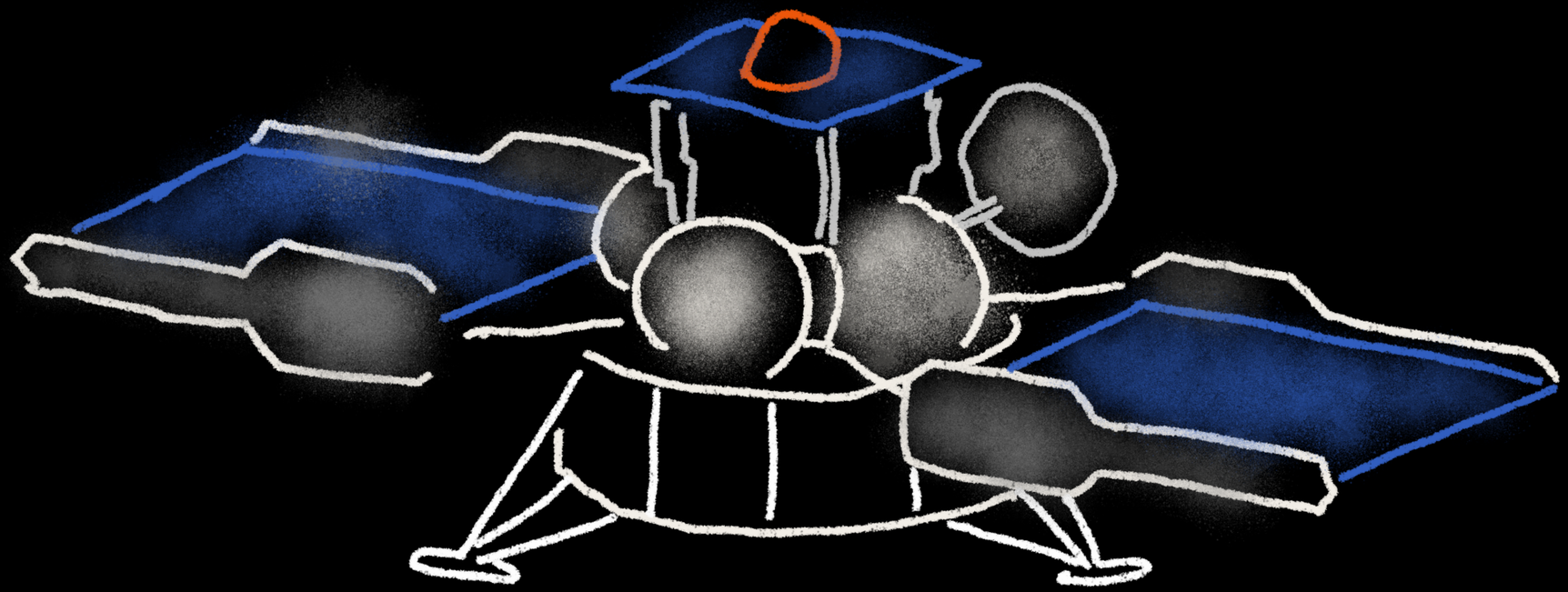
a good build radiator



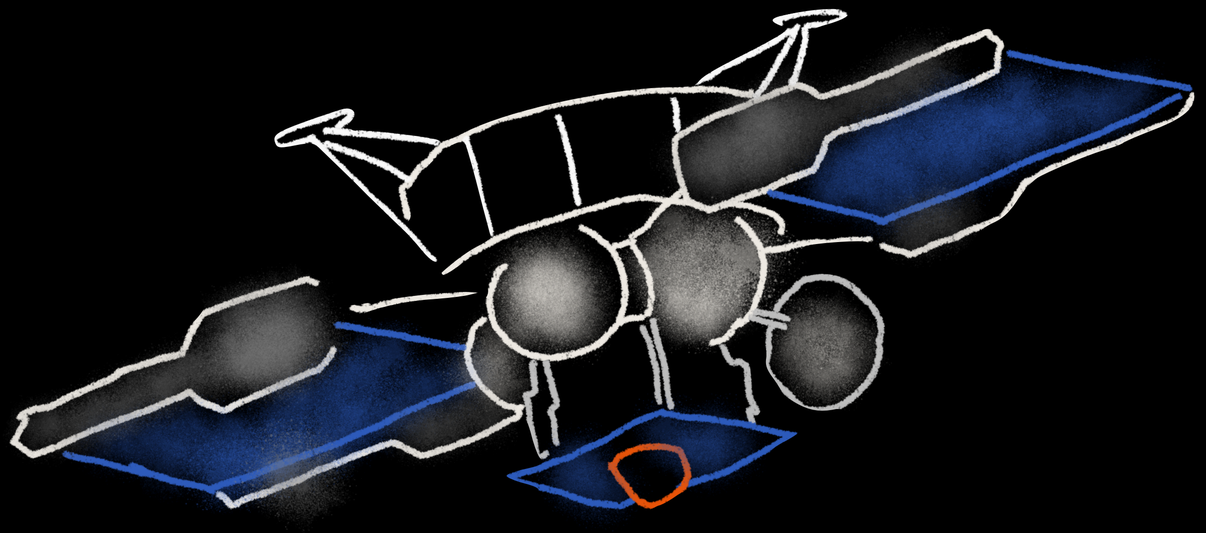


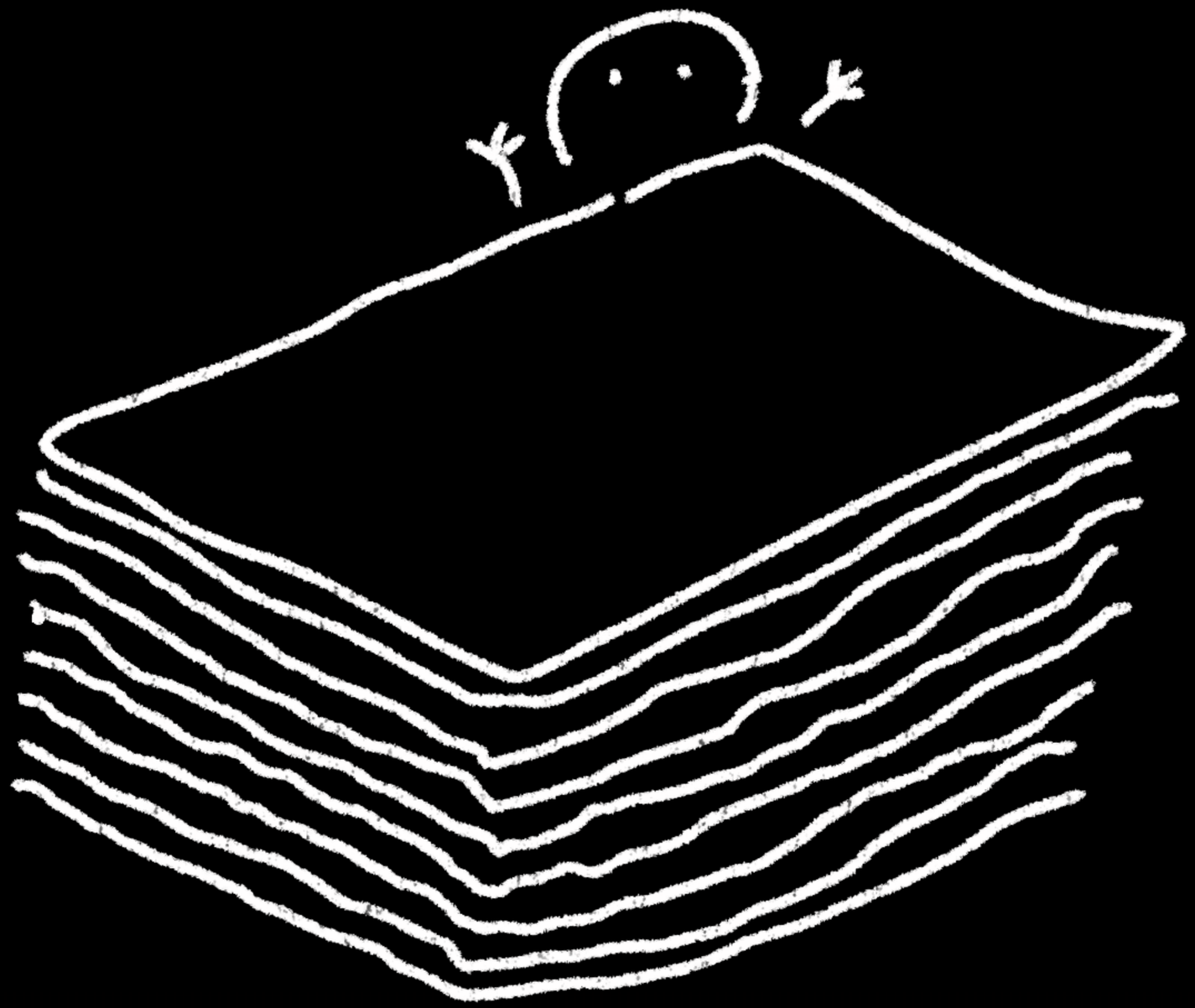
“oh yes, that build has been broken for a few weeks...”

how to brick a spaceprobe



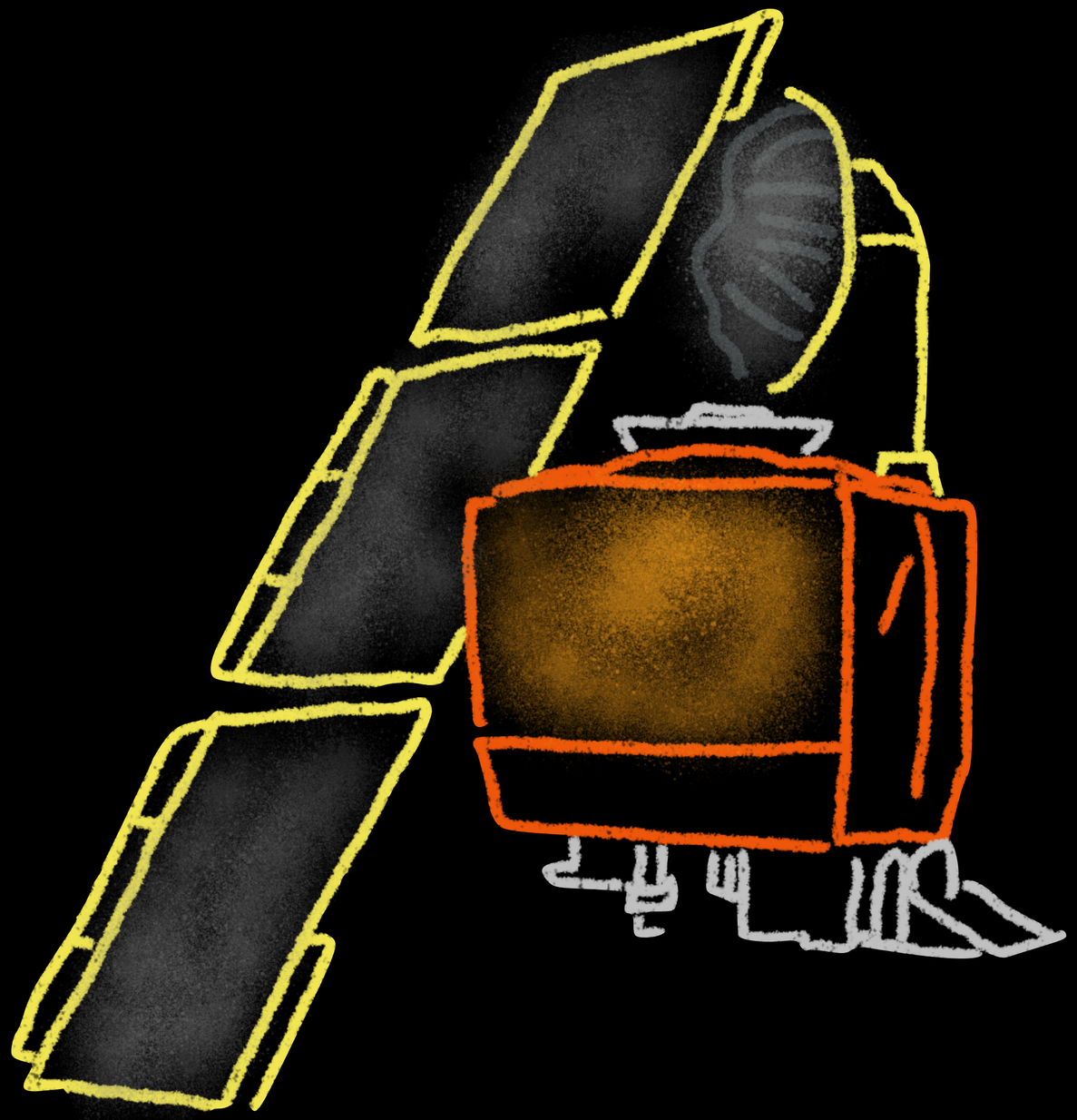
“we couldn’t get the automated checks to work, so we bypassed them”



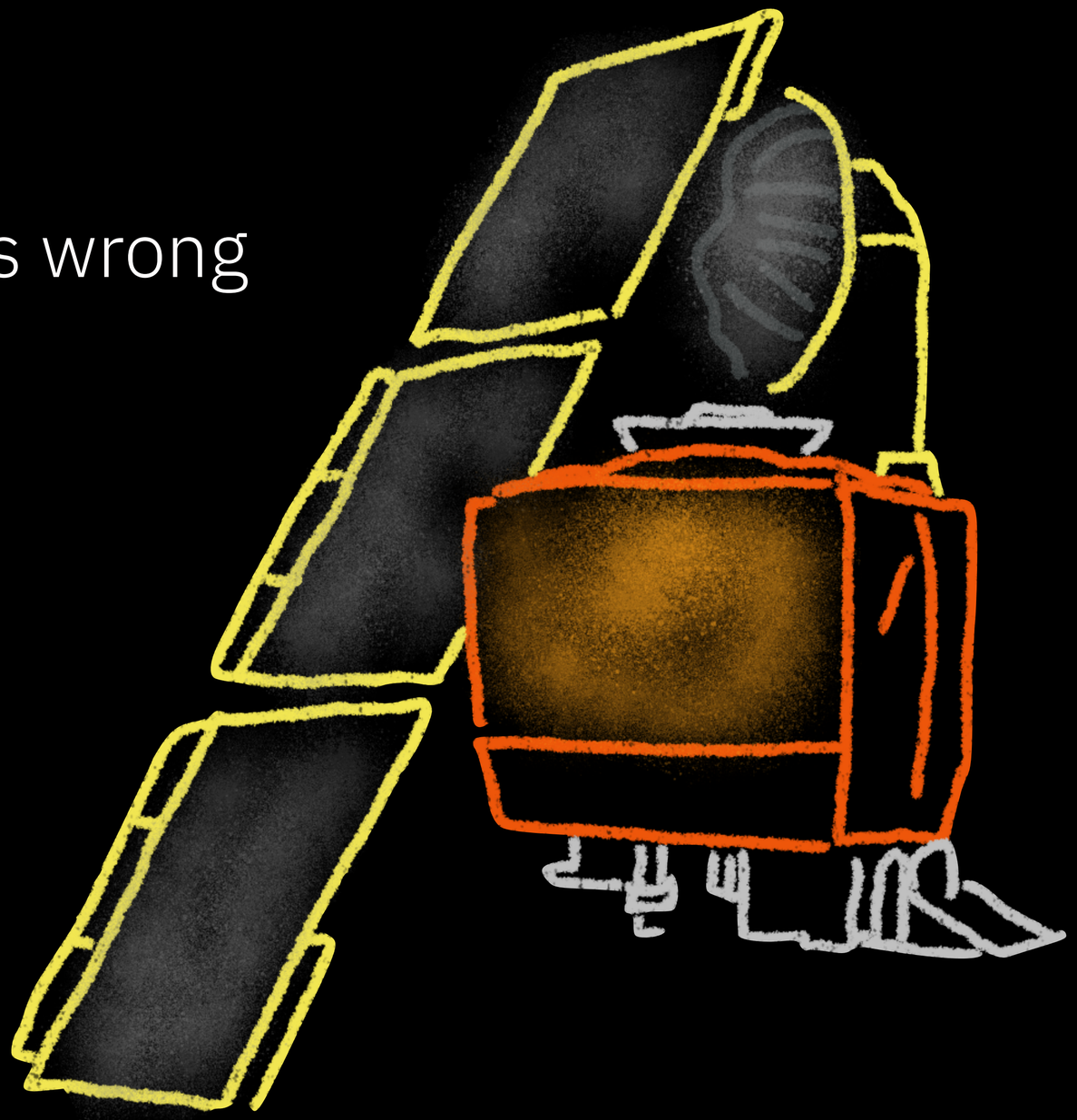


“we’ve scheduled the architecture board review for a month after the project ships”

@holly_cummins

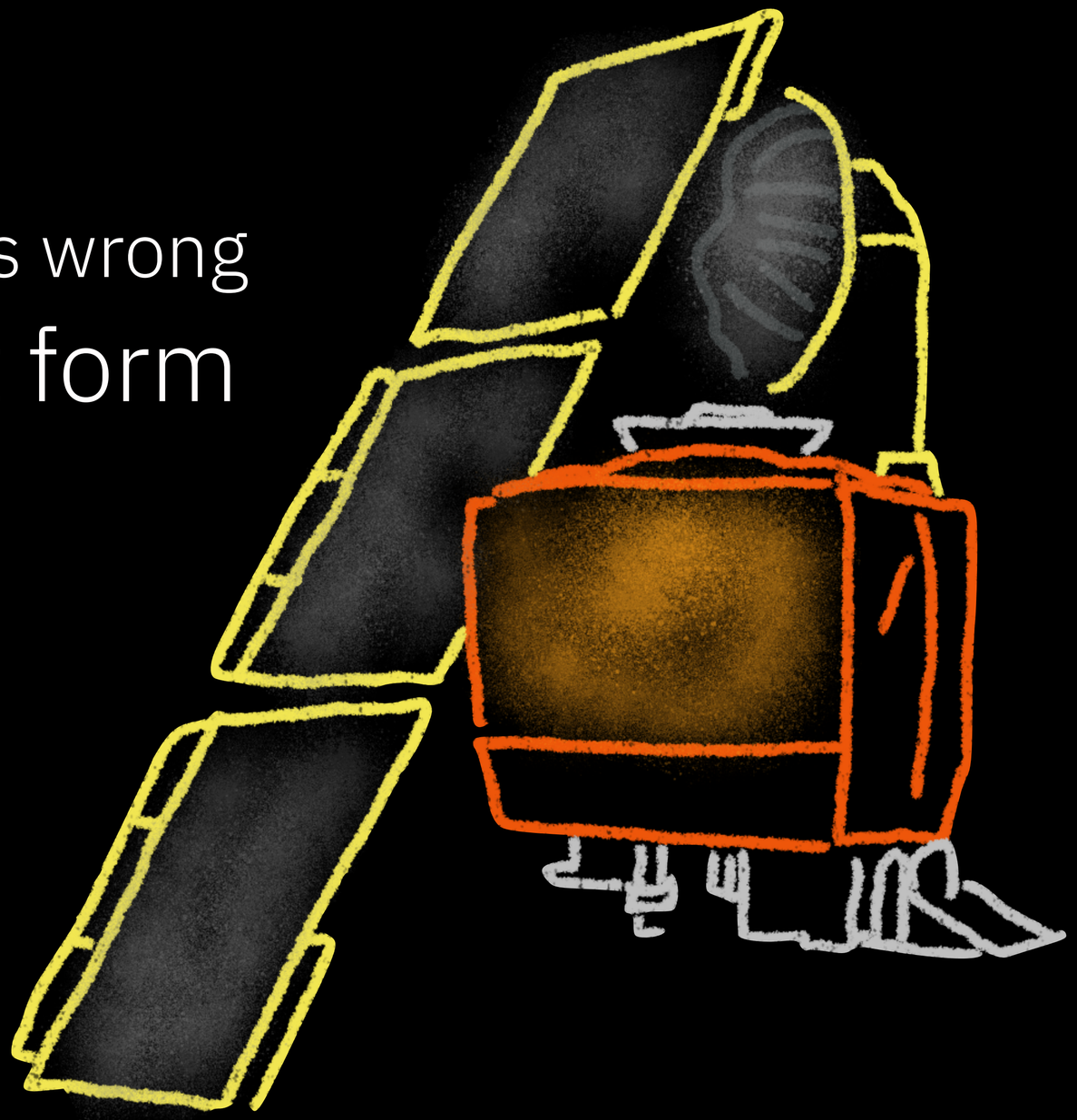


navigators warned something was wrong



navigators warned something was wrong

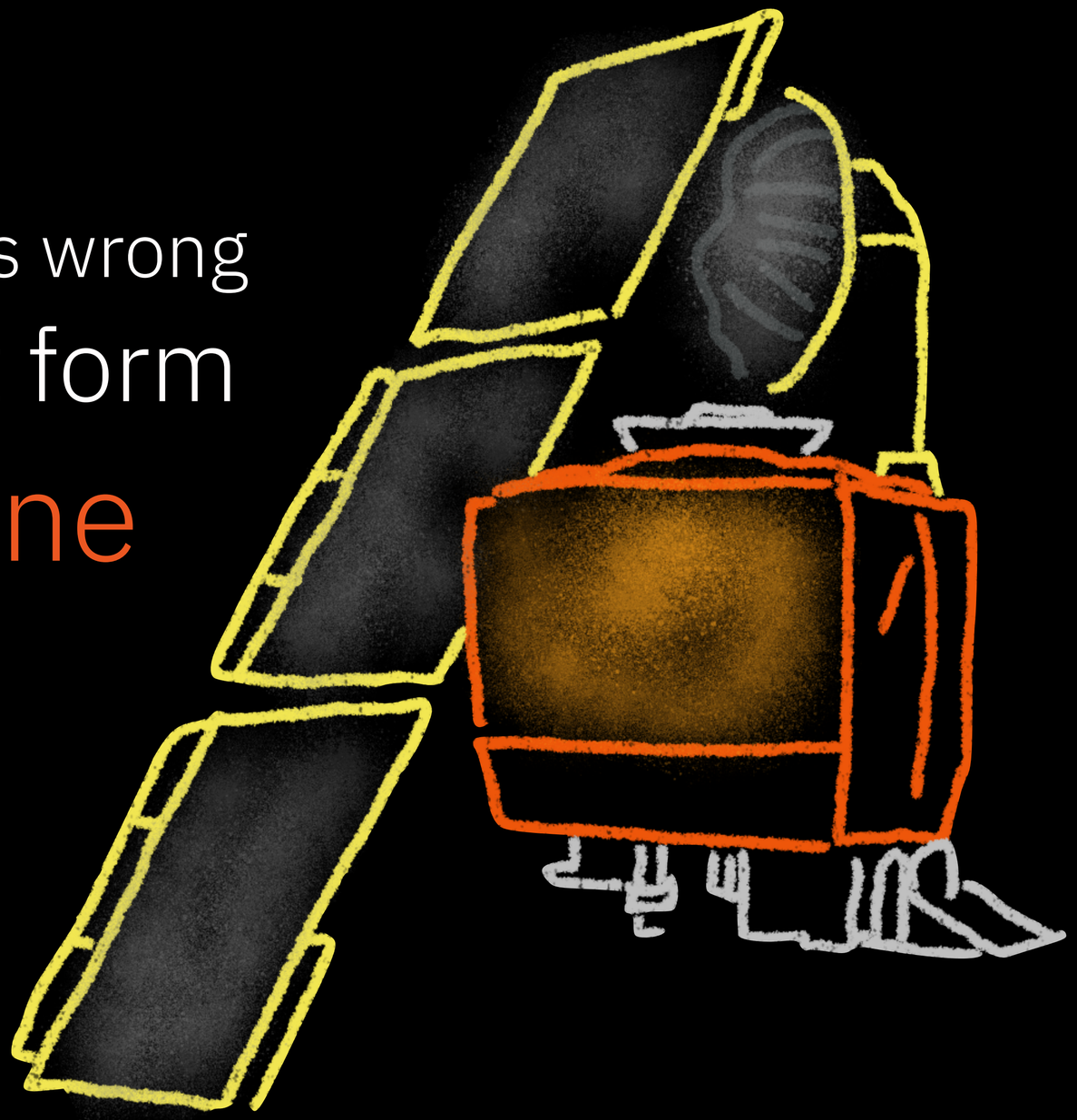
they didn’t fill in the right form



navigators warned something was wrong

they didn’t fill in the right form

so nothing was done



does the process add value?

extreme programming is the right kind of rigour

extreme programming is the right kind of rigour

test-driven development

extreme programming is the right kind of rigour

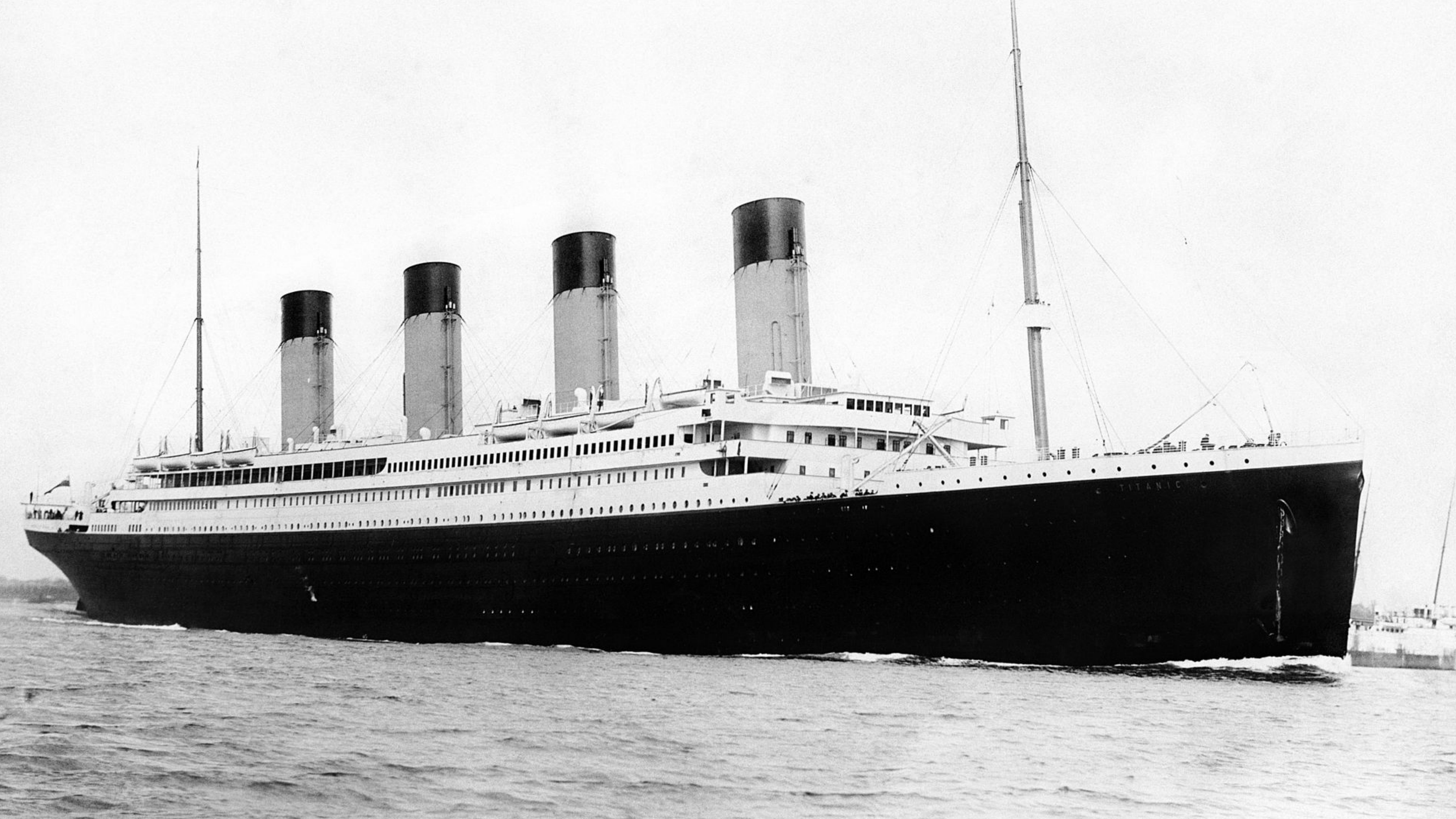
test-driven development
pair programming

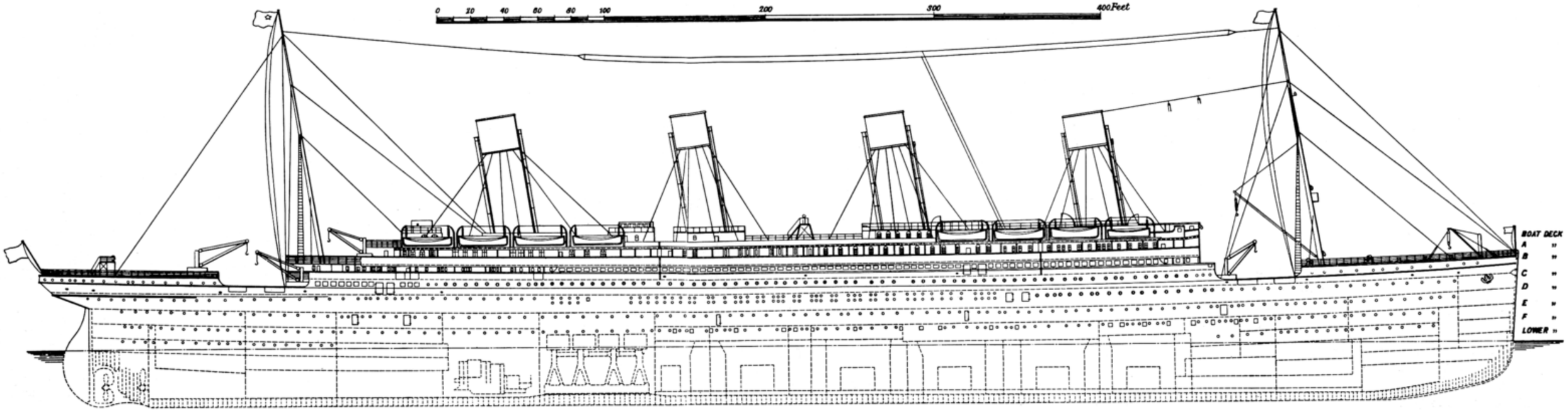
extreme programming is the right kind of rigour

test-driven development
pair programming
optimise for feedback

“but it’s in the plan”

“but it’s not in the plan”


TITANIC



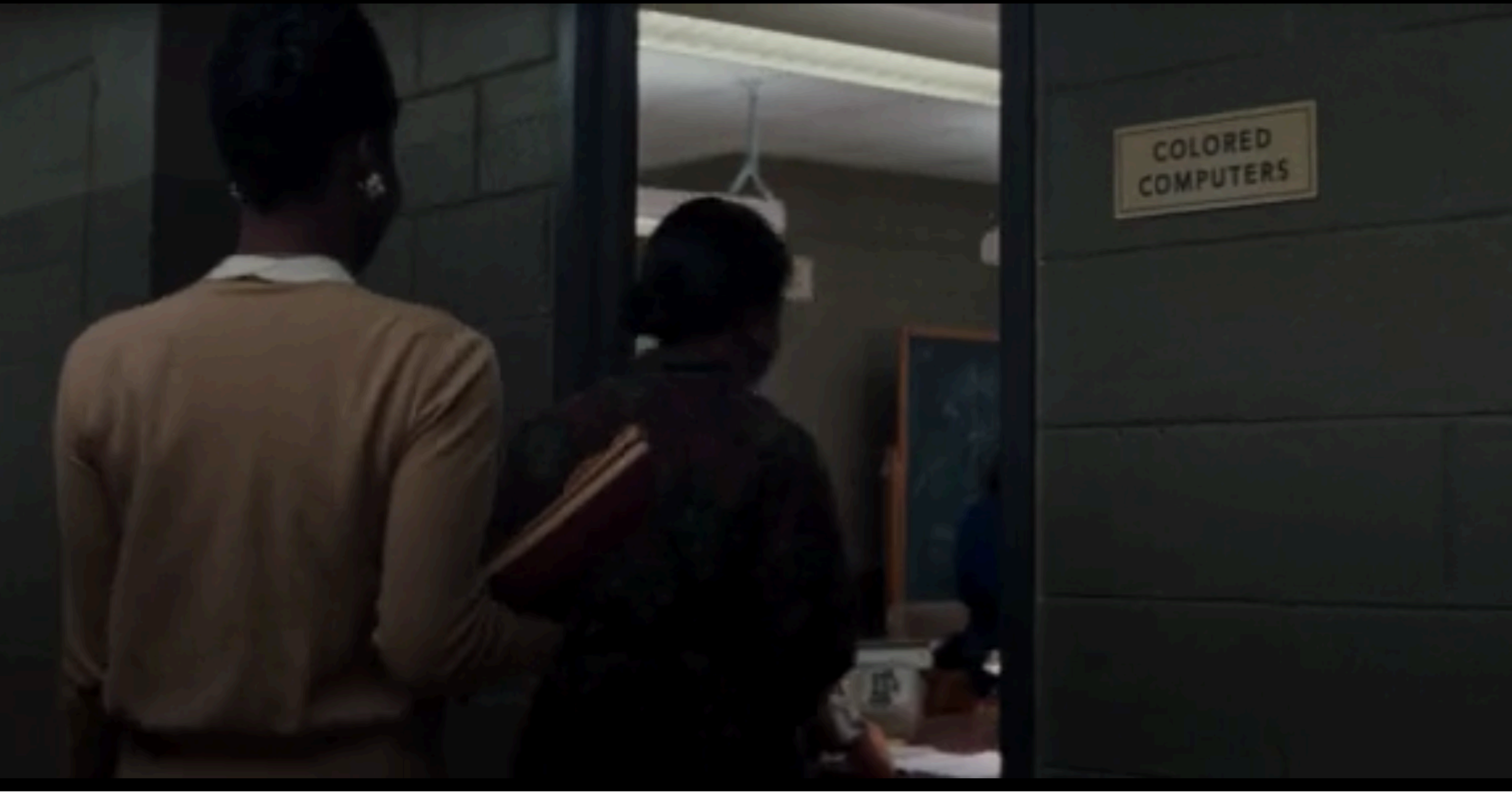
lots of bulkheads

it was too big

lookouts **saw** the iceberg

but the ship wasn’t nimble enough to avoid it

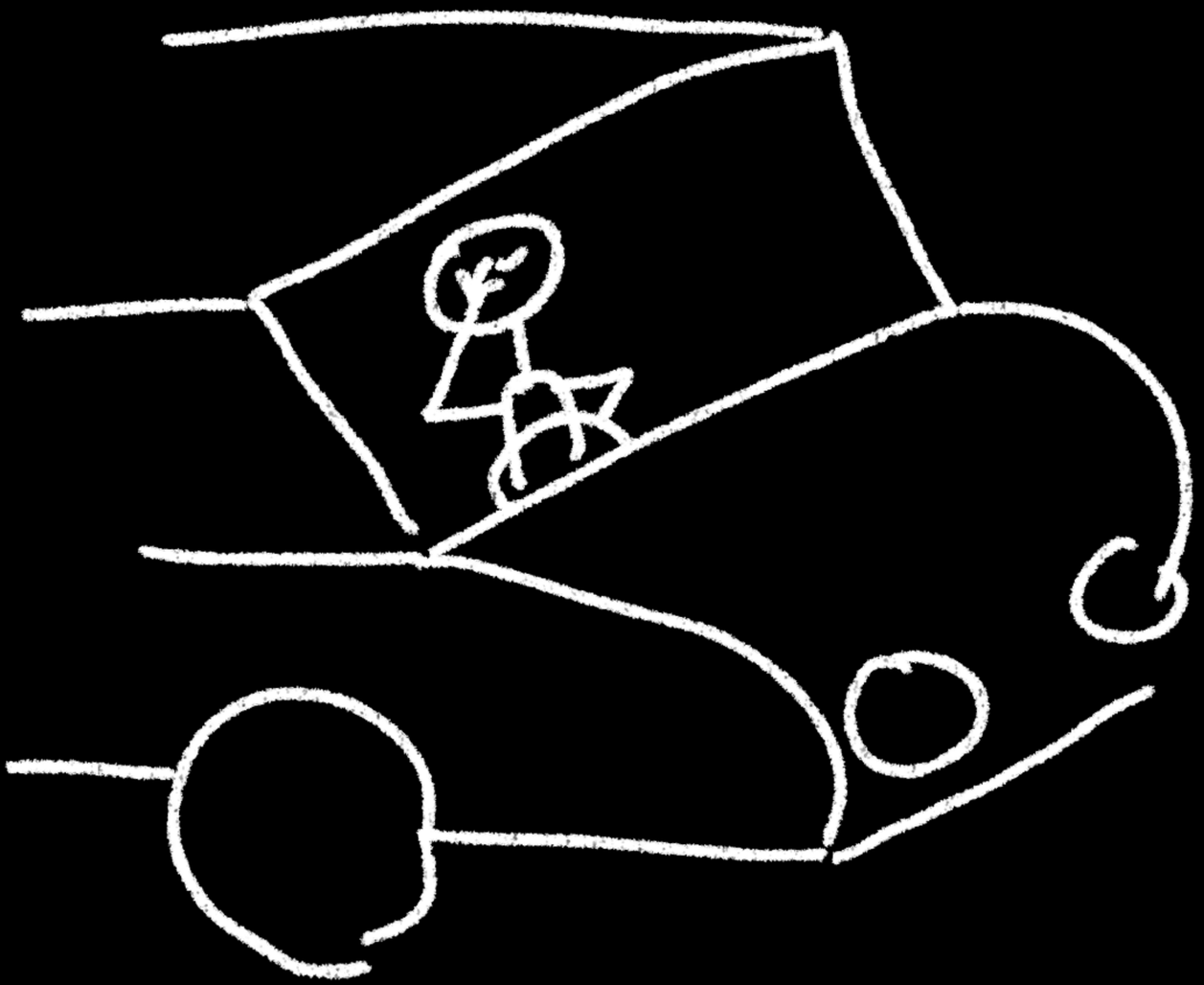
BASED ON THE UNTOLD TRUE STORY
HIDDEN
FIGURES


COLORED
COMPUTERS

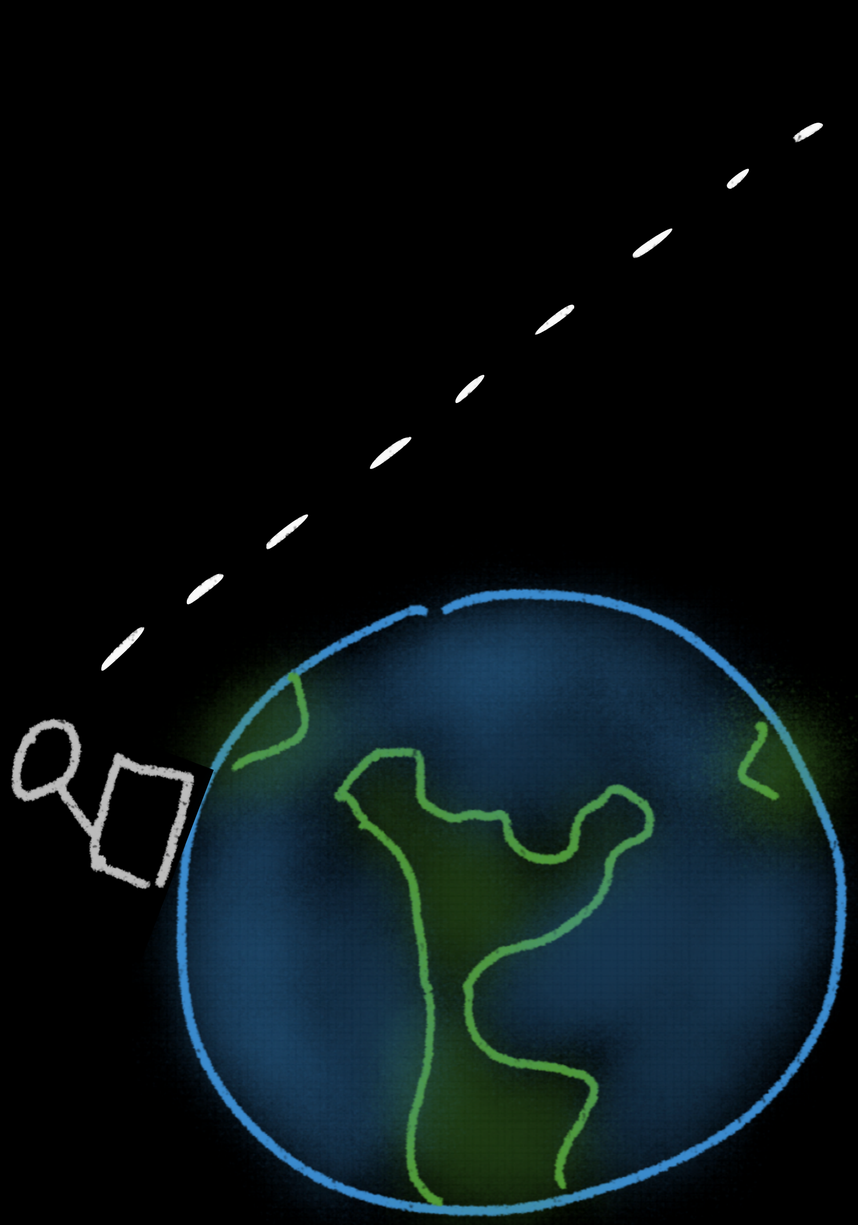
plans are **always** wrong

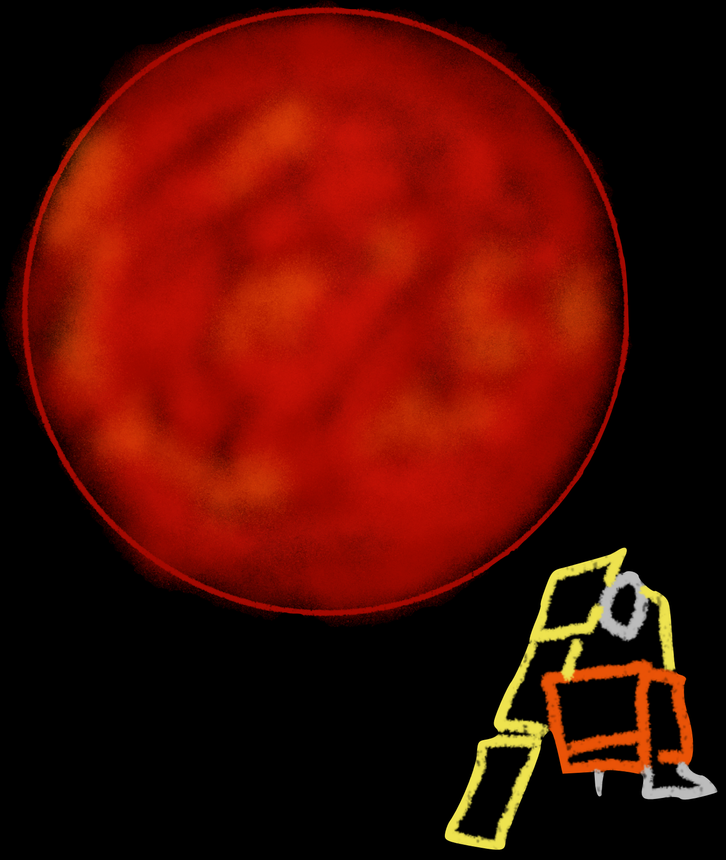
success is in how you respond

“we can’t ship until every feature is complete”



how **not** to drive a car





they often couldn’t
see the explorer

feedback is good business

feedback is good engineering

an mvp hurts

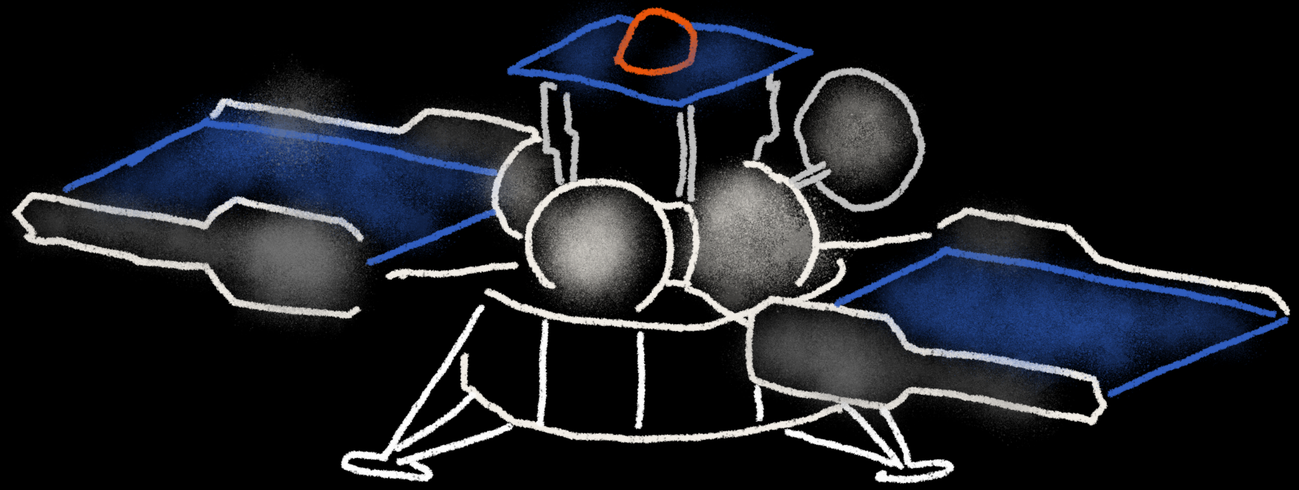
if you're not embarrassed
by your first release it
was too late

experiments can mean failure

experiments mean failure

users will have
weird behaviours

optimise for recovery

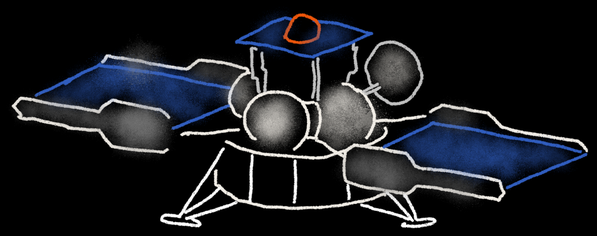


unrecoverable

back in ms
no data loss

manual
intervention

fast, but
data lost



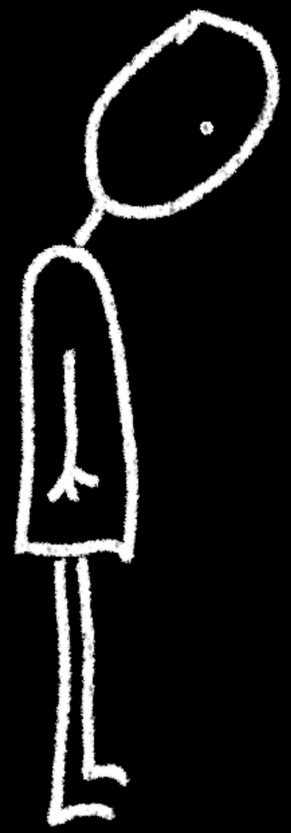
bricked

handoffs

business recoverability

remember, users will
have **weird** behaviours

speed



slow is demoralising for teams

fast is good business

A late change in requirements is a competitive advantage.

–Mary Poppendieck

more feedback → more accuracy

cloud rescued
developers from tedium

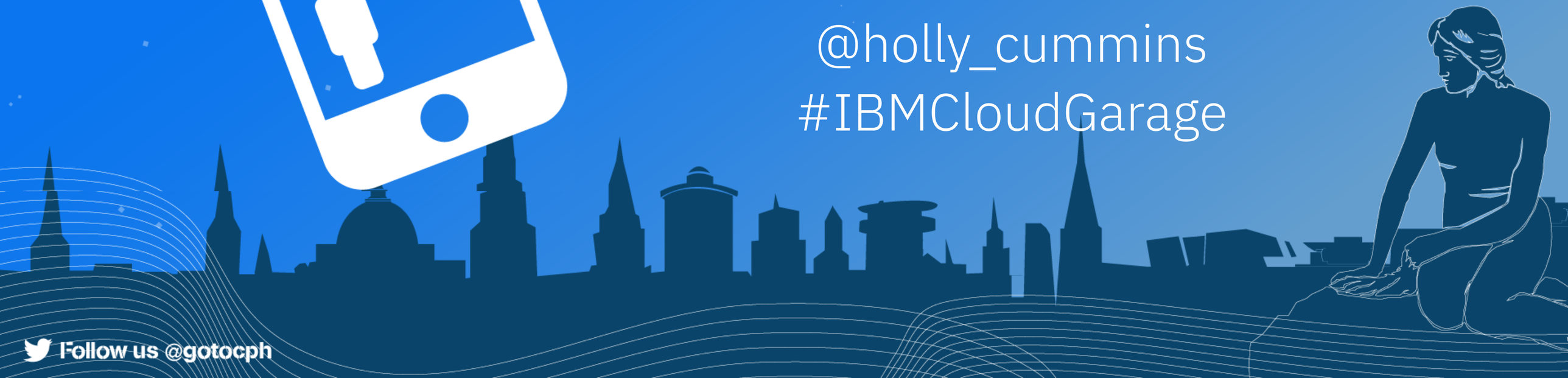
cloud native
should feel
fun

goto; copenhagen

please remember to rate this session

thank you

@holly_cummins
#IBMCloudGarage



Follow us @gotocph