
Free Fish Aren't
Free

Before we start....

Please be sure that your lines are MUTED (we'll have time for Q&A at the end and we can hear your awesome voice then)

After we're done… if you *liked* or *hated* what you saw… PLEASE fill out the session survey

Next year we hopefully can talk more at the ***Exceptional Reception!***

Agenda

1. Who?
2. Wat?
3. Why?
4. How?

1. Who?

- CRob, n, adj, and v
  - Pronunciation: U.S. (K-robe)
- Over 25 years of Enterprise-class Leadership, Architecture, Engineering, Operations, and Security experience
- Ambassador of Red Hat Product Security
- Participant in the FIRST PSIRT SIG, VulnCoord SIG, OpenSSF, and others
- Co-Author FIRST PSIRT Services Framework & PSIRT Maturity Framework
- Herds cats while doing stuff and things
- Pirate-enthusiast & hat-owner

2.
WAT?

What's up with all this THE Open Source?

Many Roles

You might write software for internal use

You might write or compose software for use by your customers

You might support software/infrastructure in your environment

You might have clicked the wrong link and am not seeing “Love after Lockup” (look it up) right now

Pop Quiz! (yay!)

A 2019 Synopsis report states that somewhere between 60-80% of modern software is comprised of OSS (yay!)[1]

Who here knows they use OSS in their Enterprises?

Who thinks they don't have ANY OSS on their network?

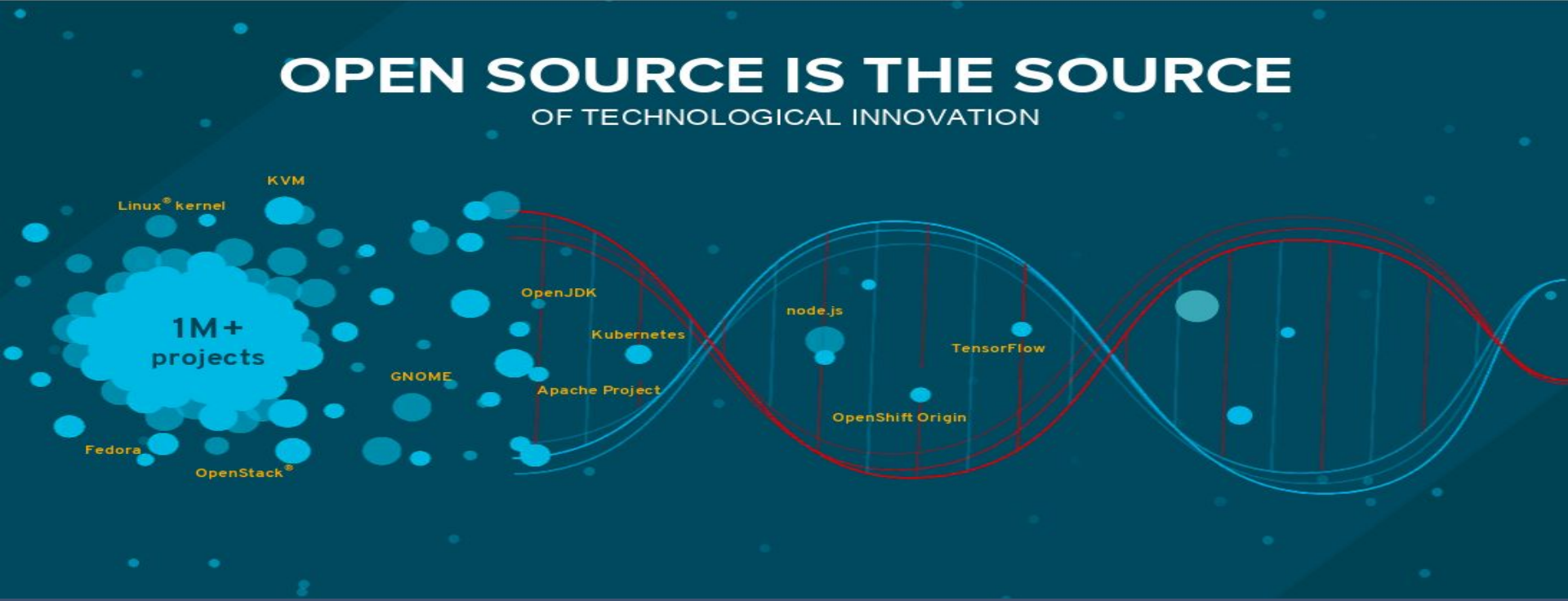
[1] - https://www.linuxfoundation.org/wp-content/uploads/2020/02/oss_supply_chain_security.pdf

Why?

Speed
Agility
Innovation
Powerful ability to configure for-purpose
Penguins
Awesome black and green screens

Open Source “won”

Yay us!



“Blah, Blah, Blah, Marketing slide, Blah, Innovation, Blah, plus tax”
- CRob

https://octoverse.github.com/ - GitHub reports over 96mil projects with active utilization in 2018

"open source" refers to something people can modify and share because its design is publicly accessible.[1]

It's FREE, so it MUST be good, right?

1 - https://opensource.com/resources/what-open-source

CLOSED SOURCE

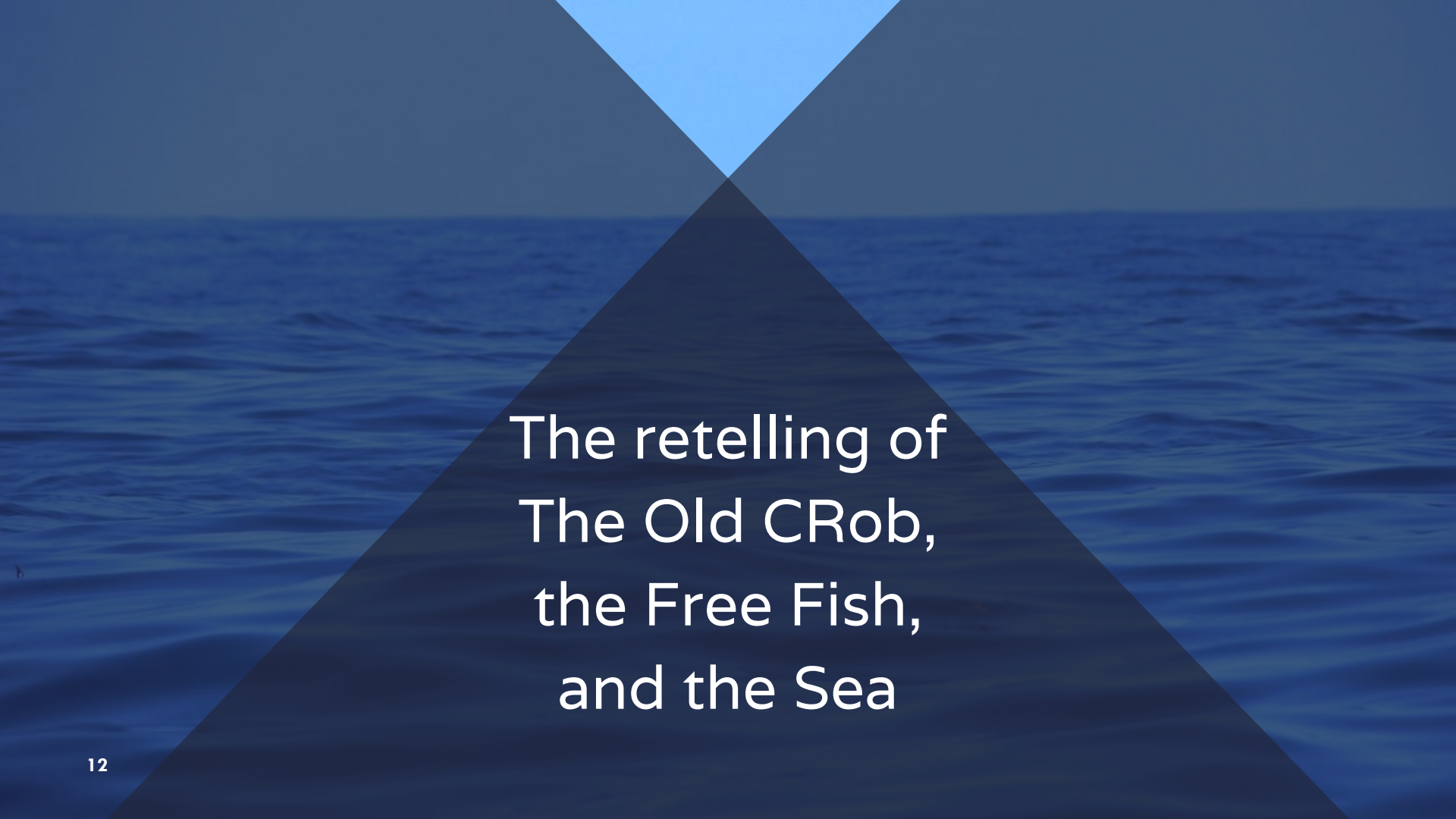
- “Traditional” Software.
- Probably single-supplier.
- Unknown practices to create, package, & deploy products (might have great docs to share)
- “One throat to choke” for support, updates, etc
- Might be embedding/using OSS….

Think: “Waterfall”

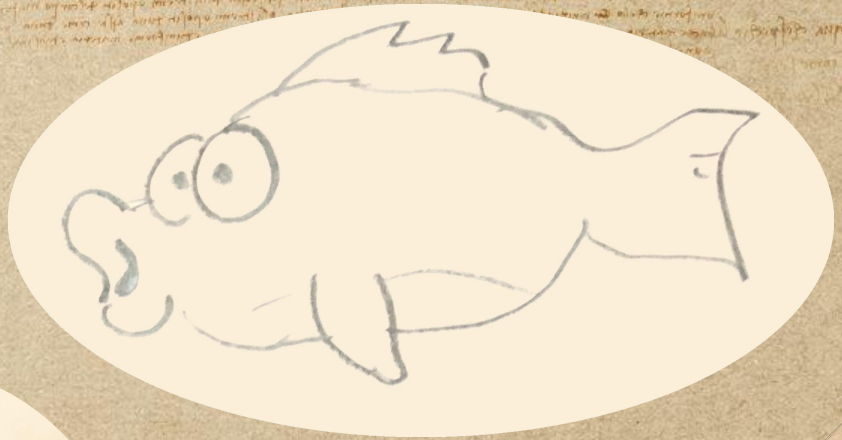
OPEN SOURCE

- “New” software model with different processes/practices
- Multiple contributors
- Open and auditable processes and code (that YOU can go audit)
- Rarely a single source for support and updates
- Depending on size/maturity of “team” varying levels of quality

Think: “Agile”, but with stronger opinions

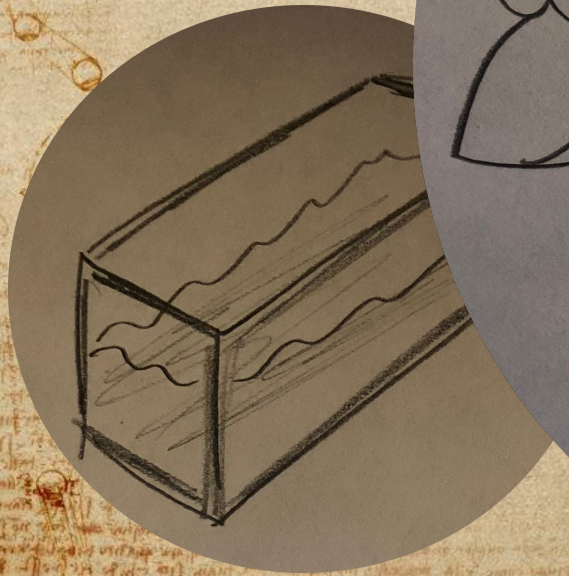

The retelling of
The Old CRob,
the Free Fish,
and the Sea

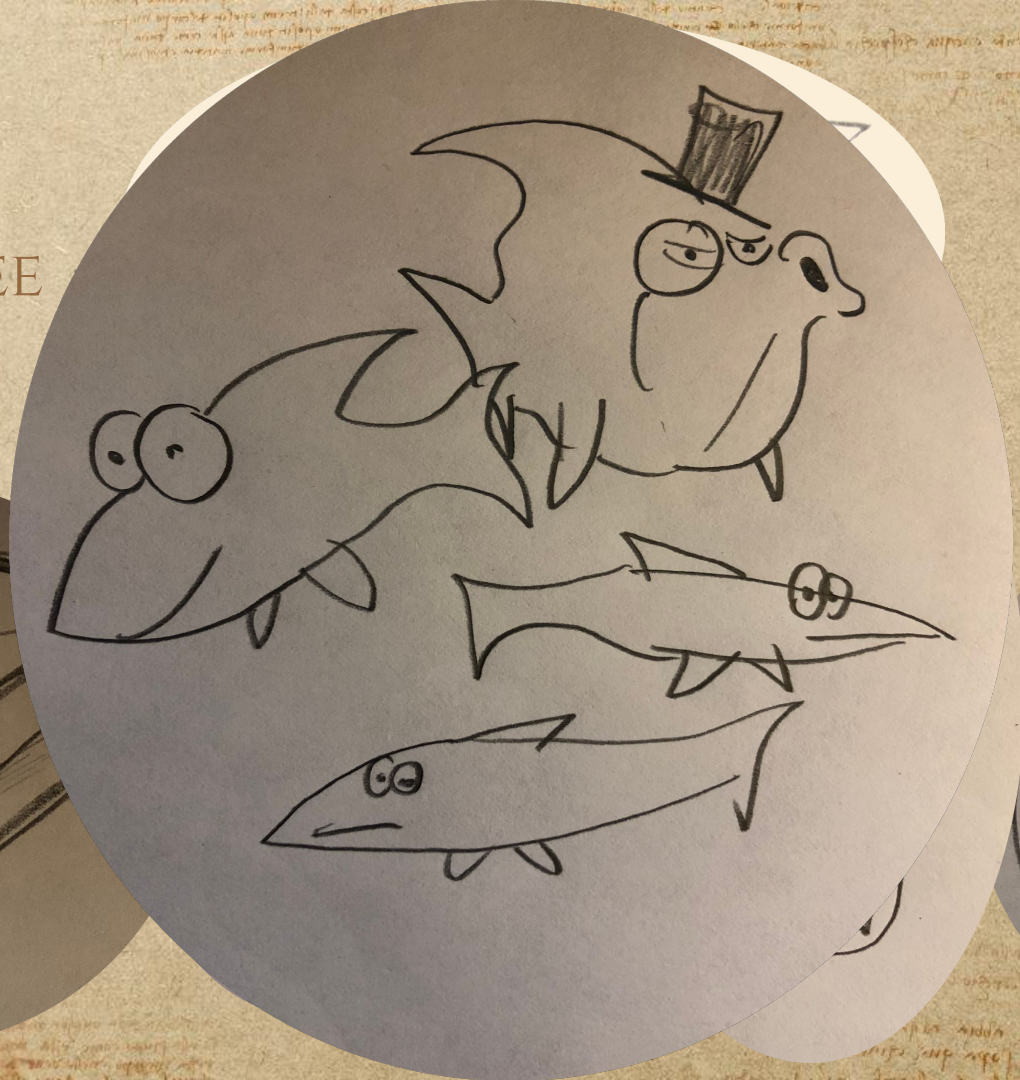
....It all started one bright and beautiful day with a simple 5th grade Science experiment

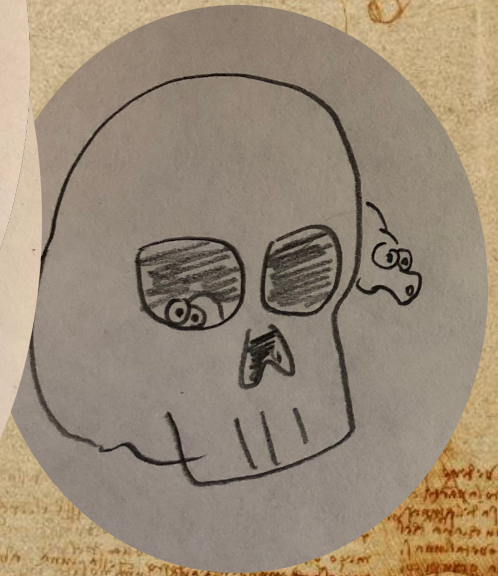


CROB'S "FREE FISH" BILL:

>$200 USD







The moral

Open Source is kinda like a free fish

You generally have extra work to do around the fish...

It can be pretty cool (that bubbly pirate chest was THA awesomes)!

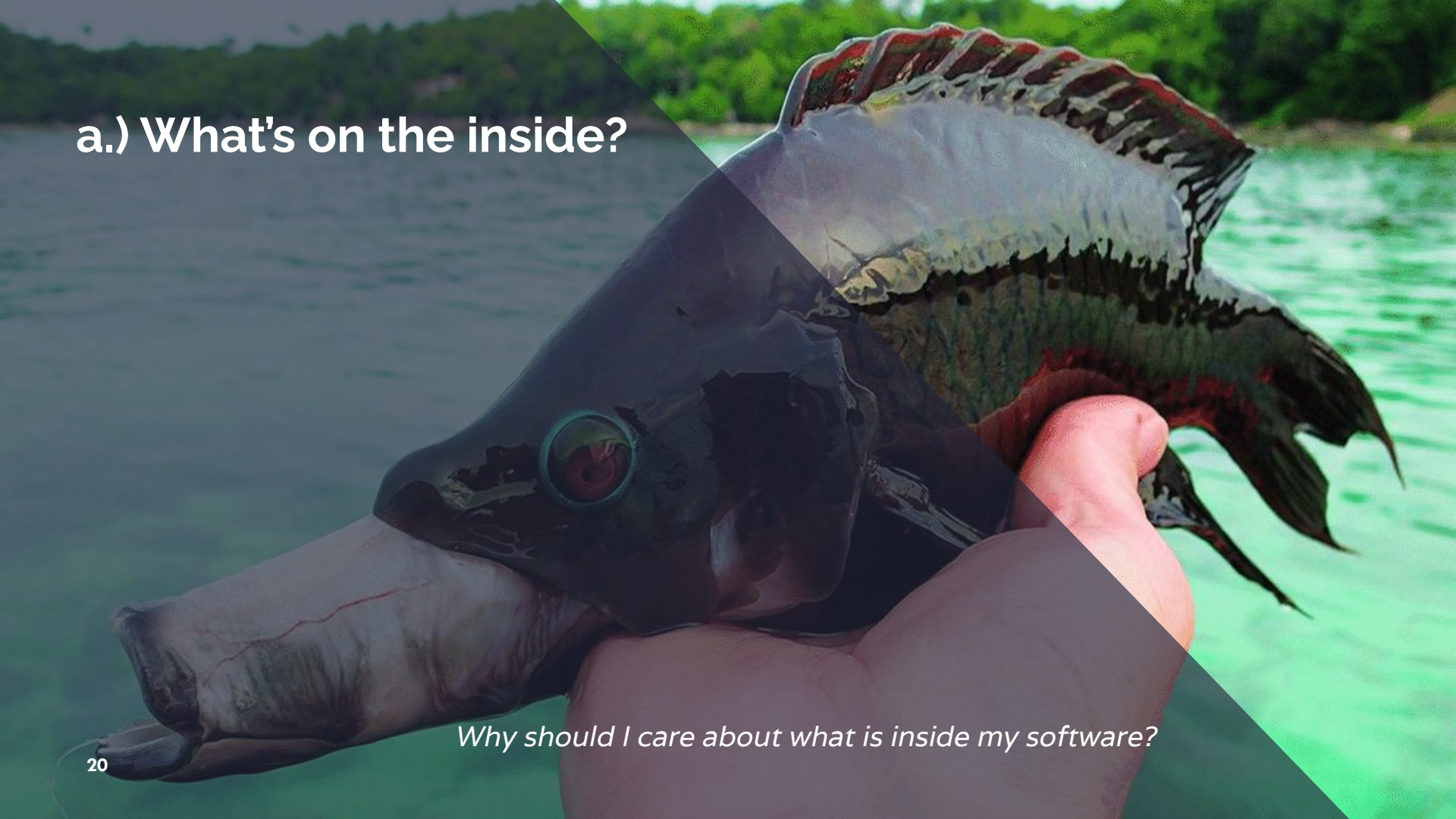
If you leave a 12 yr old to oversee it it's GOING to die.
<--FACT

.... In other words, with a little bit of effort, it can be beautiful, but it takes work!

3.
WHY?

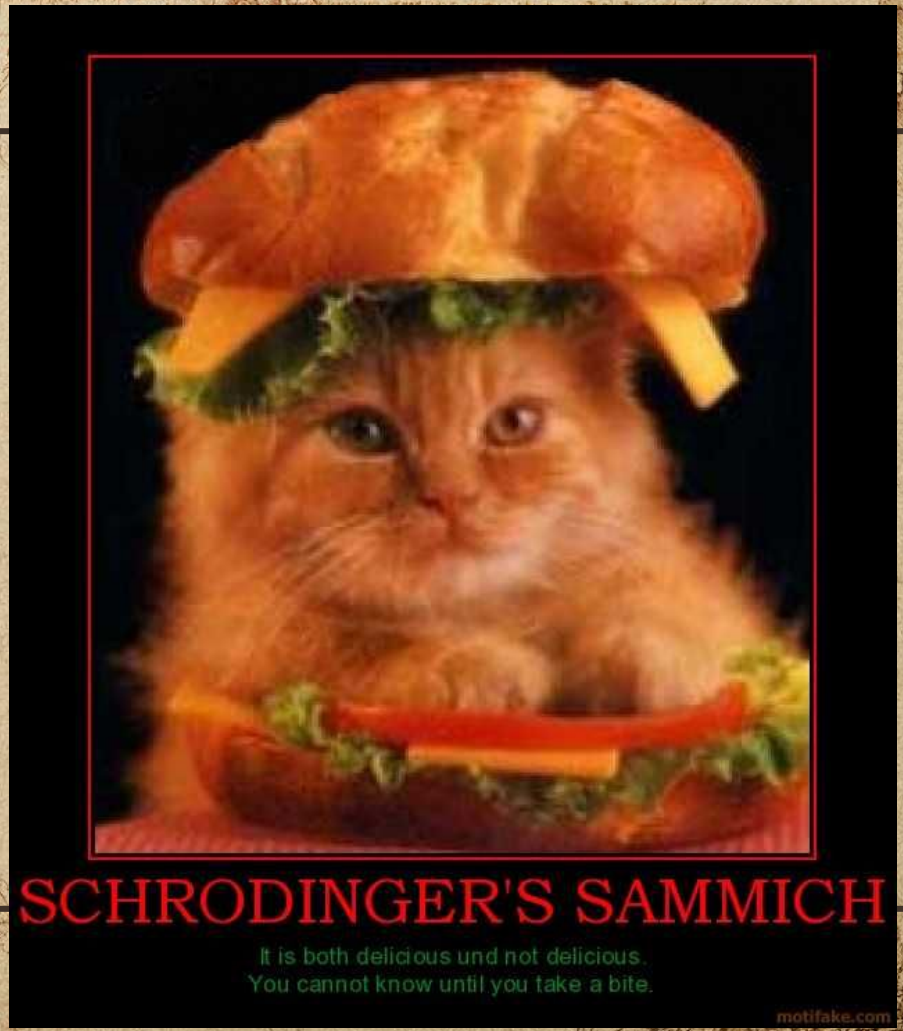
Why it is important to know MOAR about OSS (aside from the obvious coolness of it all!)

a.) What's on the inside?



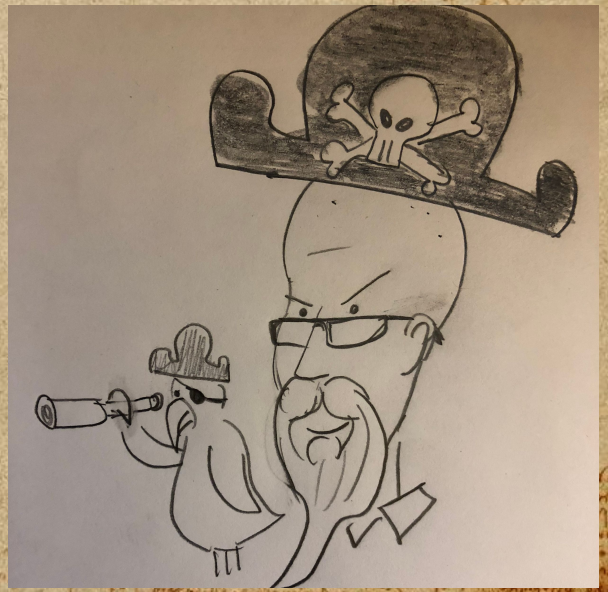
Why should I care about what is inside my software?

The Sandwich Paradox


SCHRODINGER'S SAMMICH
It is both delicious und not delicious.
You cannot know until you take a bite.
motifake.com

What's in the sammich and how it was made matter

- Chain of custody/package ancestry matter
- Trusted builds by trusted maintainers on protected infrastructure
- Changes documented and tracked
- Signed with authorized_keys



How do you ***know*** what's in that package you just installed?

b.) Who made it and how is it going to be supported?

Does it REALLY matter who made my software or where it comes from?

RIPPED! From the headlines.... (zomg!)

https://www.infosecurity-magazine.com/news/open-source-supply-chain-attacks/

How are YOU going to monitor and support it?

Some fish food for thought:

How was the code made and tested?

How do they vet new code submissions?

How often are updates provided?

How are those fixes communicated? Do they do advisories?

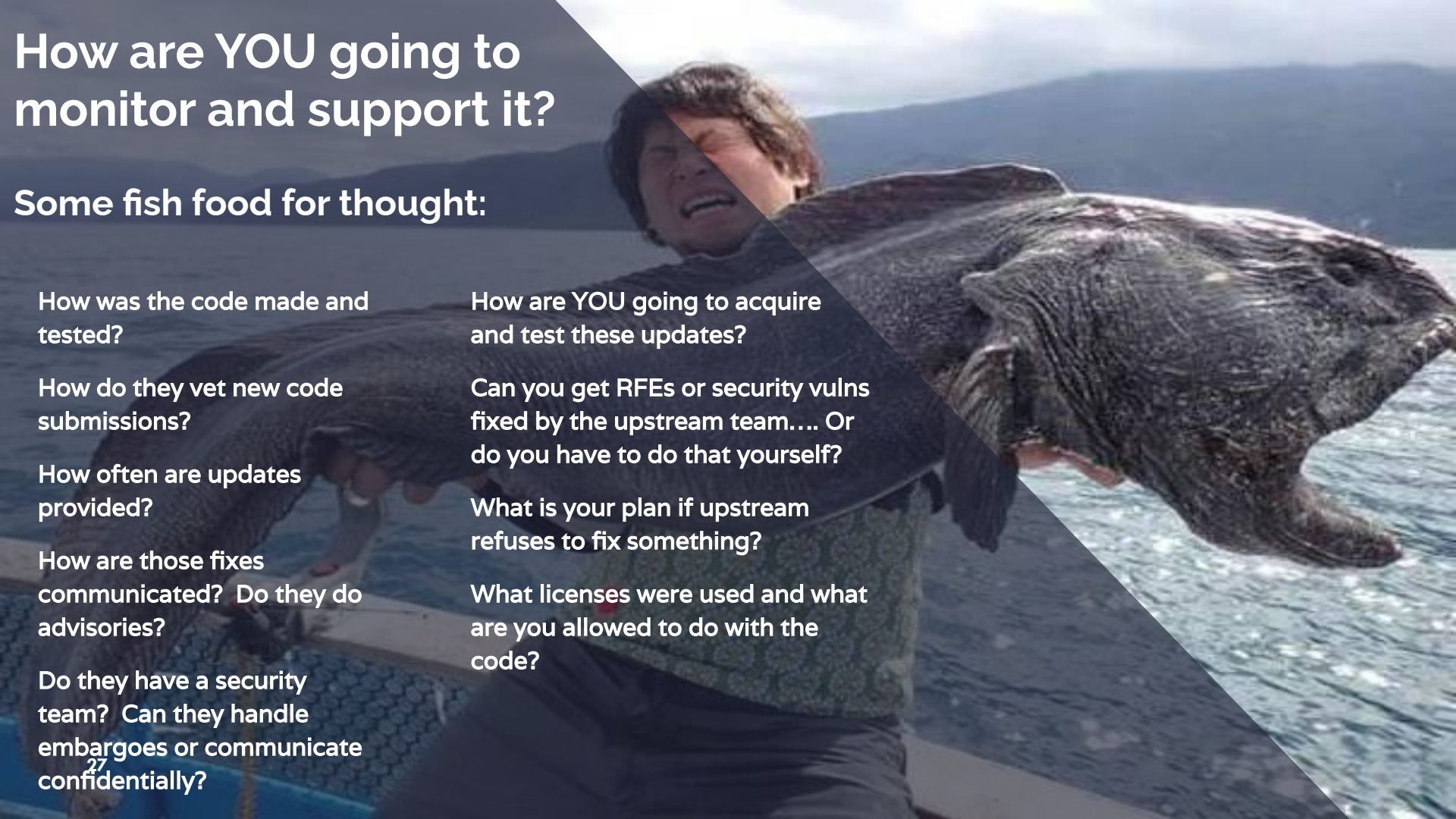
Do they have a security team? Can they handle embargoes or communicate confidentially?

How are YOU going to acquire and test these updates?

Can you get RFEs or security vulns fixed by the upstream team…. Or do you have to do that yourself?

What is your plan if upstream refuses to fix something?

What licenses were used and what are you allowed to do with the code?



Open Source comes in many flavours

Aka - there is no **THE** Open Source

Project

Smaller efforts, class projects, “for fun”, POCs.

You should expect little to no/”best effort” help from these

Community

More formalized. A group of folks working towards a more defined goal.

Will have their own rules/culture

Commercial

Someone is getting paid to support the software.

You have support terms & contracts

It may be shocking to find out, but even the mighty OSS has some potential risks that need managed…..

Risk? Wut Risk?

Uncontrollable upstream

Unknown Contributors

Immature project

Change in Technical direction

Abandoned Project

Lack of Security team or mindset

Unknown contents

4.
HOW?

How can you better MANAGE your OSS-usage?

DIY

Commercial tools

Monitor “github” commits

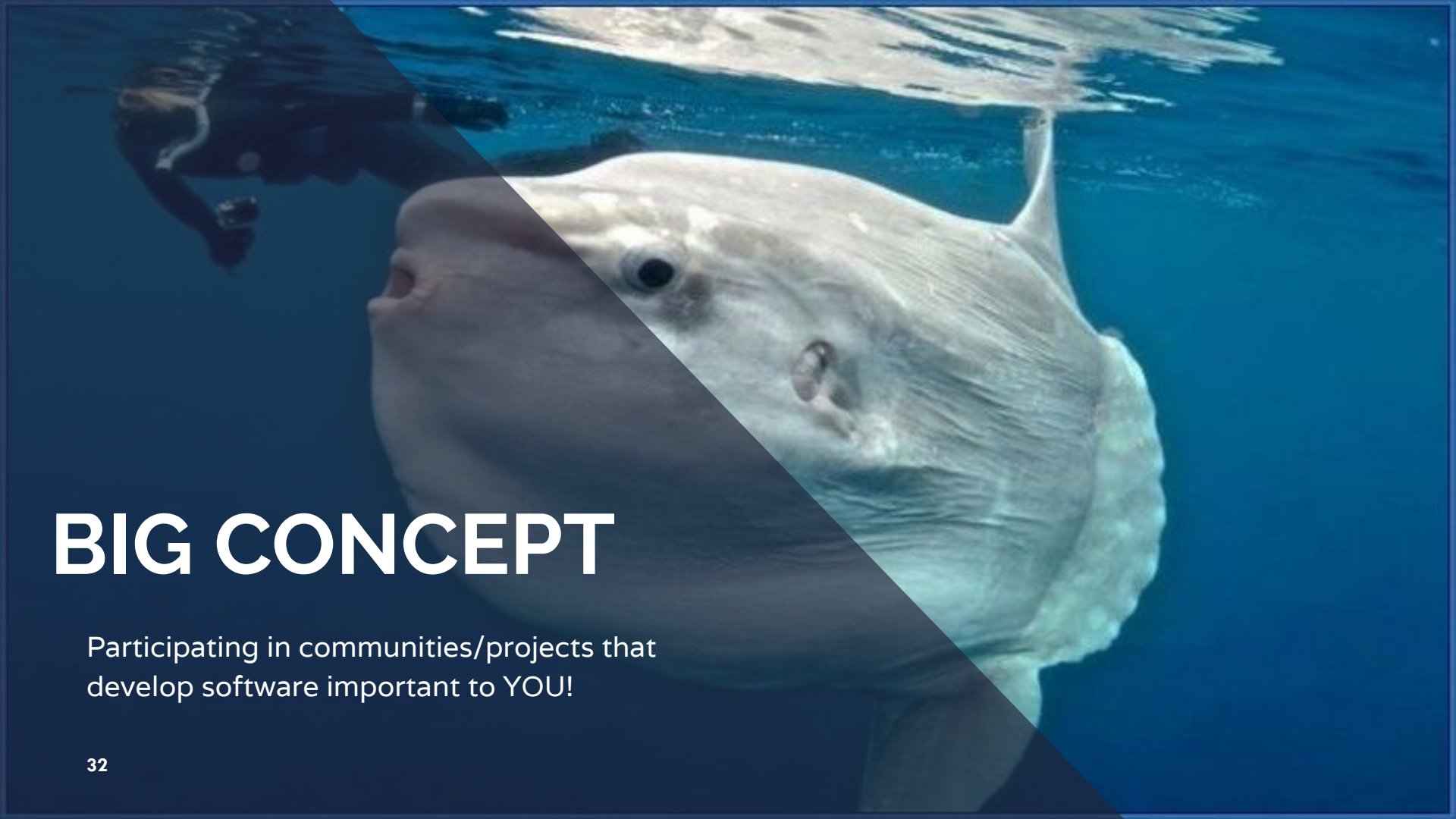
Mailing lists

Change logs

NVD*

Bug Bounty Vendors

Many of these require a VERY thorough understanding of the packages you're consuming and their dependencies



BIG CONCEPT

Participating in communities/projects that develop software important to YOU!

BIG CONCEPT

...OR... find a friend to help you out.

Commercial OSS Support - *Free software for a Fee*

Your mileage may vary, but in general, what you should expect -

Trusted/reproducible builds

Regression/security testing

Contracts to support you

License indemnification

A Product Security Team

You have A LOT of options, educate yourselves on what is best for YOUR business and Risk appetites

Free-ish Tools & Tricks

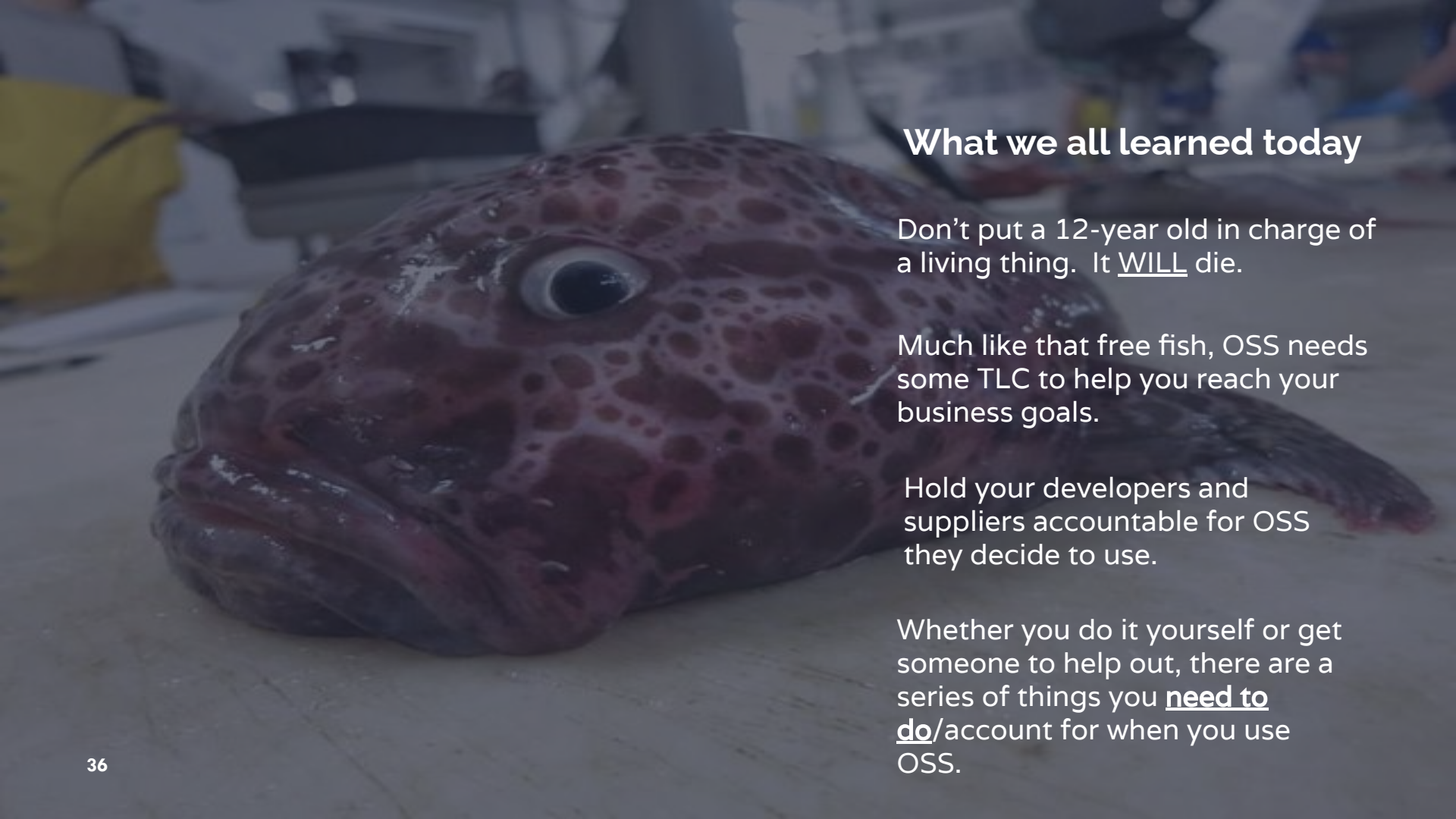
GitHub Stuffs
https://github.com/features/security
- Vuln Scanner
- Dependabot
- Secret Scanning
- …

OWASP Dependency Checker (SCA Tool)
https://owasp.org/www-project-dependency-check/

ZAP (Webapp Attack Proxy)
https://owasp.org/www-project-zap/

Linux Foundation CII Best Practices Badges
https://bestpractices.coreinfrastructure.org/en

- Wherever possible automate ALL THE THINGS
- Depending on your SCM tooling/build infrastructure, leverage built-in or bolt-on tools
- Only choose software from known, good sources



What we all learned today

Don't put a 12-year old in charge of a living thing. It WILL die.

Much like that free fish, OSS needs some TLC to help you reach your business goals.

Hold your developers and suppliers accountable for OSS they decide to use.

Whether you do it yourself or get someone to help out, there are a series of things you **need to do**/account for when you use OSS.

Questions

Free Fish Aren't Free

Thanks!

CRob_at_RedHat_dot_com

@RedHatCRob