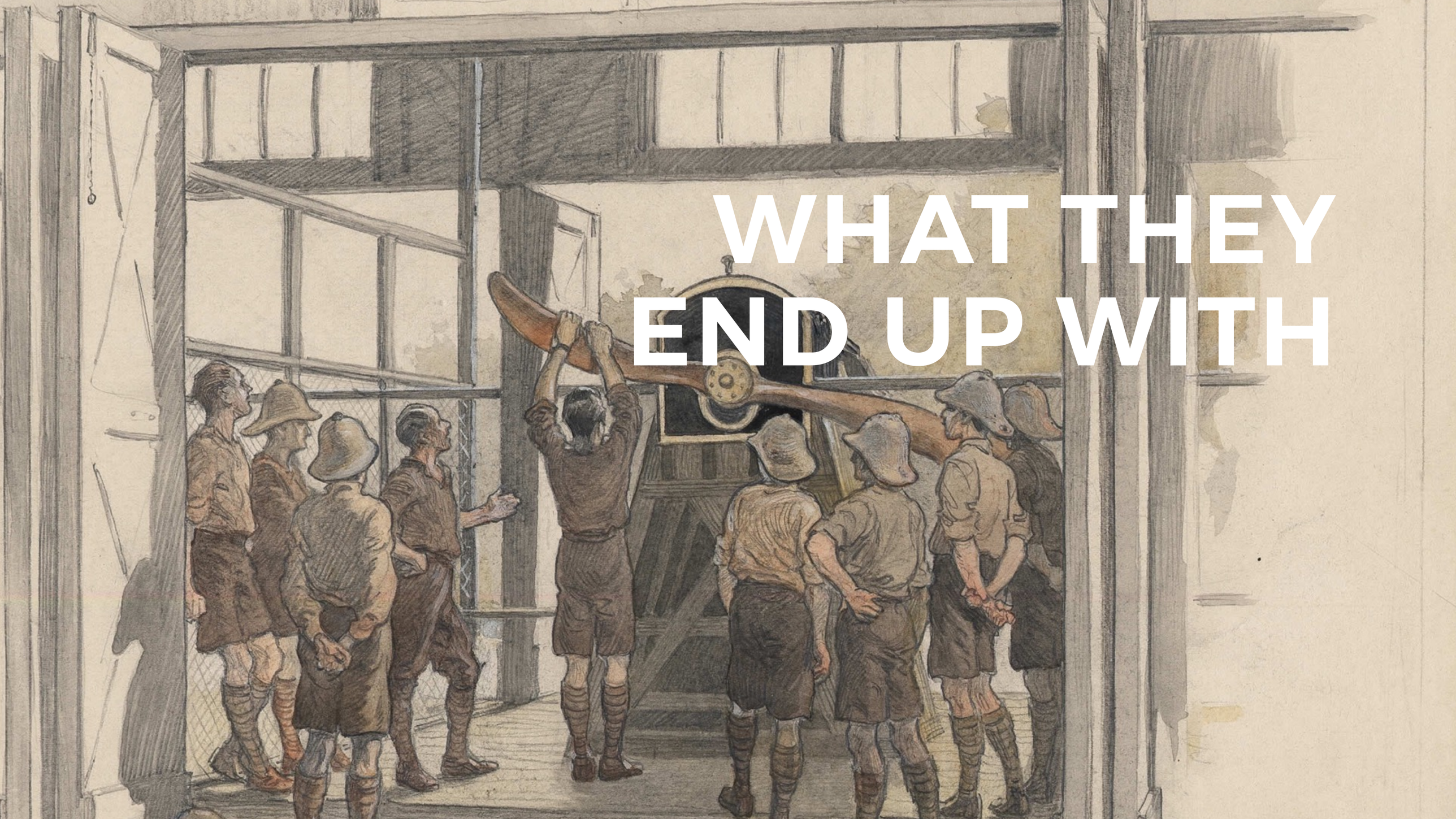
HOW THINGS ARE
OR COULD BE
tooling, learning, and culture
Upp

WHAT IS ADVERTISED



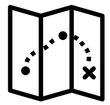
WHAT THEY END UP WITH

OUR TOOLS

Quickcheck, PropEr, Triq

Property-based Testing

Probably the best of its kind available across all languages and communities



Live Debugging

Tracing

Any part of the system can be introspected at run-time, local or distributed, and it keeps improving.

Dialyzer

Optional Type checking

Erlang lacks some powers from static languages, but benefits greatly from some solid analysis.

CutEr, Concuerror

Even more testing tools

Research projects keep delivering amazing tools like CutEr and Concuerror

The language itself

The Erlang model and OTP

Can't ignore this being a thing everyone benefits from. Not much to add there!

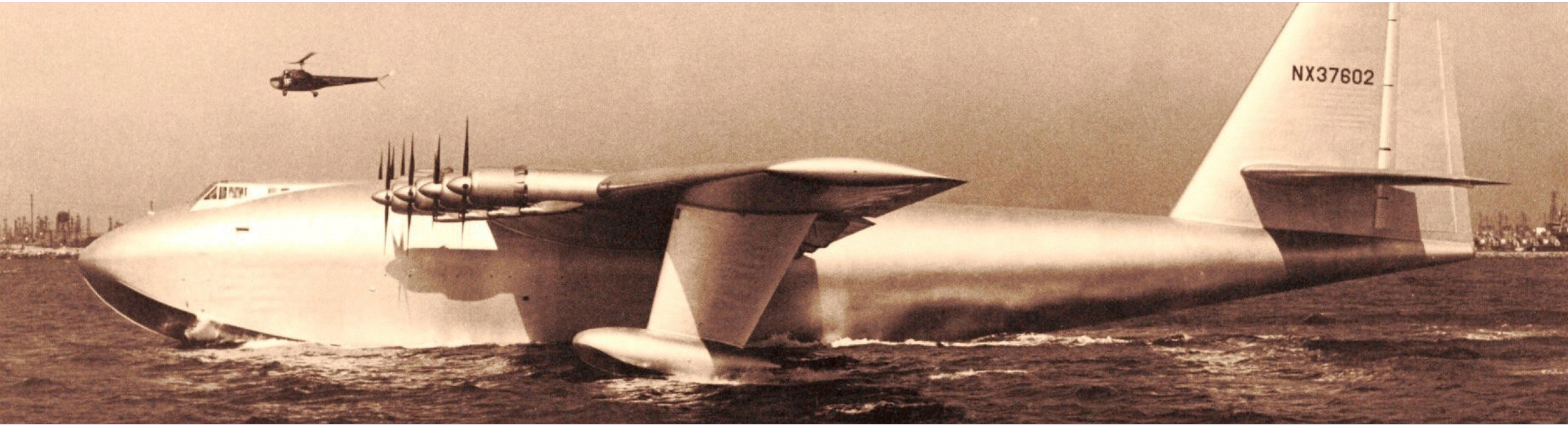
Too many to name them all

Libraries of all kind

An amazing number of Erlang libraries are floating around the ecosystem.

County Rd
Gambia St
A4289
A4289
A4289
The Magic Roundabout
Shrivenham Rd
Shrivenham Rd
Queens Dr
A4312
A4312
Queens Dr
Queens Drive Post Office
MoneyGram (inside Post Office - Queens Drive)
Esso
Fleming Way
Fleming Way
Fleming Way
B4289
B4289
Wiltshire Fire & Rescue Service
The Frying Fish
Swindon Fire Station

COMPOUNDED HARDSHIPS



The spruce goose is a project from WWII as an attempt to carry large cargo across the Atlantic by flight, without using Aluminium because of wartime restrictions. The resulting vessel was made mostly of birch and is officially a ‘flying boat’. It has done a single flight, remaining airborne at 21 m off the water at a speed of 217 km/h for around 1.6 km.

THE BELIEF

WE HAVE GROWN A TOLERANCE

Our experience gives us perspective

HISTORY OF ERLANG BUILD/DEPENDENCY TOOLS

1986

ERLANG CREATED

Build system unknown, probably something scary by today's standards

1990-2000s

MAKEFILES

Makefiles are the standard (only) way to do things. Dependency management is unspecified and ad-hoc

2010

REBAR

Makefiles made in Erlang. Adds dependency handling for the first time off source files.

2013

ERLANG.MK

Rebar made as Makefiles.

Improves on speed, meant to be patchable

2016

REBAR3

Actually not that similar to REBAR.

Adds package management with Hex. Stricter project structure

30 YEARS OF TOOLMAKING

Boils down to: makefiles and Erlang and back, calling down to standard Erlang command line tools, nitpicking on configuration format, fetching dependencies from git, eventually getting packages (which totally depend on work by the Elixir community)

THE PROBLEM

WE HAVE BECOME INSENSITIVE

And we wear the scars of our bad experiences as badges of honor

GARRETT SMITH

gar1t 2:14 PM ☆
time ./bootstrap

real 0m9.958s
user 0m10.476s
sys 0m1.808s

This can't be your workflow

mononcqc 2:14 PM
it is.

tristan 2:15 PM
I just use compile and/or escriptize before running tests

gar1t 2:15 PM
Do you have comics to read during that op?

mononcqc 2:15 PM
I don't recompile every other character ¯_(ツ)_/¯

you can try escriptize, it may be faster

tristan 2:15 PM
or just compile if you are just running tests

mononcqc 2:15 PM
rebar3 escriptize 0.87s user 0.20s system 81% cpu 1.317 total

Since I joined

FREQUENTLY ASKED QUESTIONS

HOW DO I RUN CODE OUT OF THE SHELL?

Most people had to share makefiles and invocations of erlc to run stuff, with manual ‘start’ commands or scripts

WHAT IS OTP?

When basic Erlang got in place, people then started trying to exploit that OTP thing, and wanted a gentler way into that world.

HOW DO I BUILD A RELEASE?

Exploring OTP, people learned about releases. There was no great way to do them, always time consuming.

FREQUENTLY ASKED QUESTIONS

HOW DO I DO RELUPS?

That's the hard test; how can we actually deliver easy, repeatable hot code loading that isn't a hack?

HOW DO I MAKE MY OWN HEX MIRROR?

I've started hearing this one more and more. Hex and package management is starting to take roots.

???

Who knows what's next!

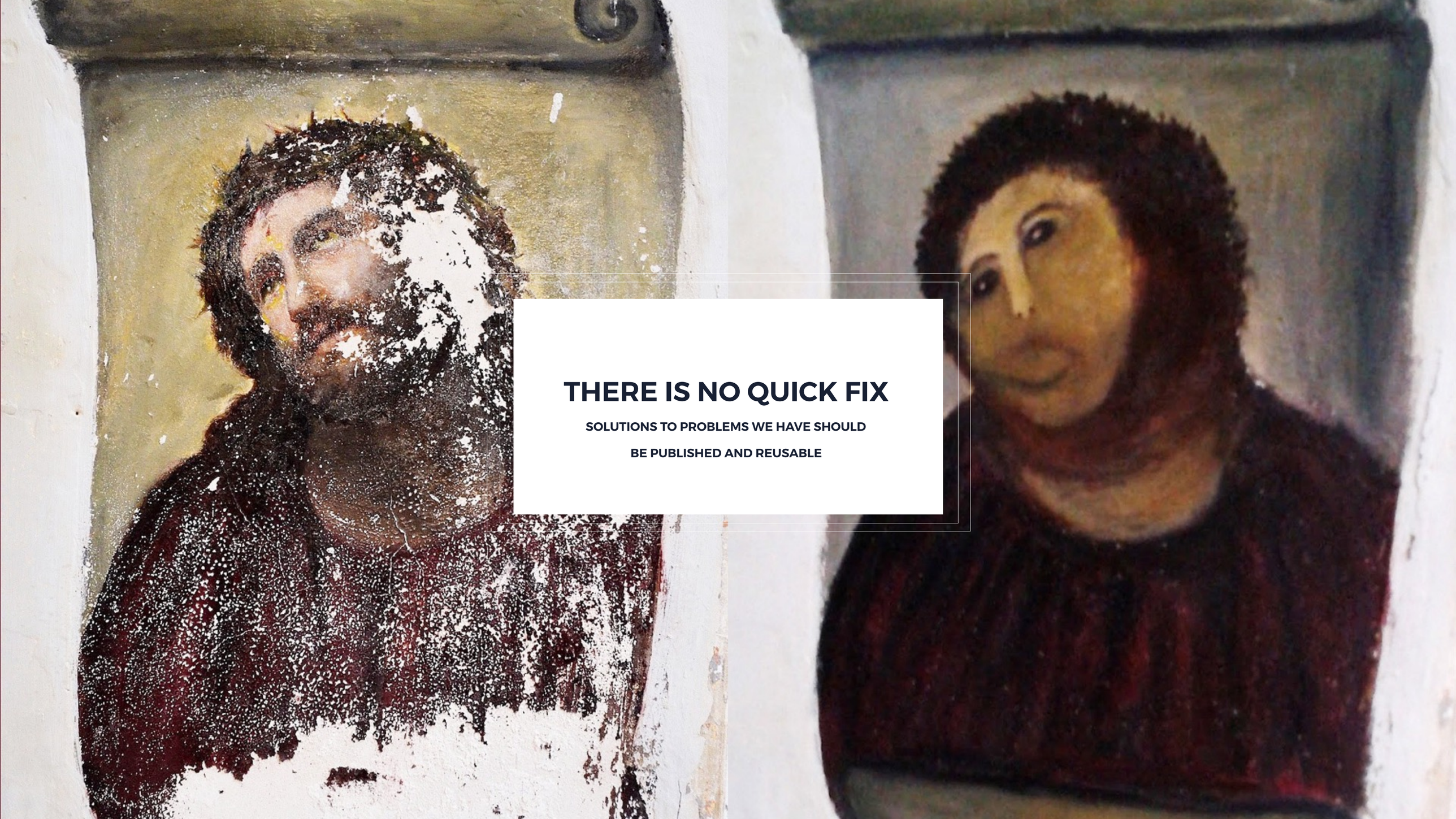
A NIGHT WITH ERLANG

A NIGHT WITH COMFORT

HOW THINGS COULD BE

HOW TO GET THERE

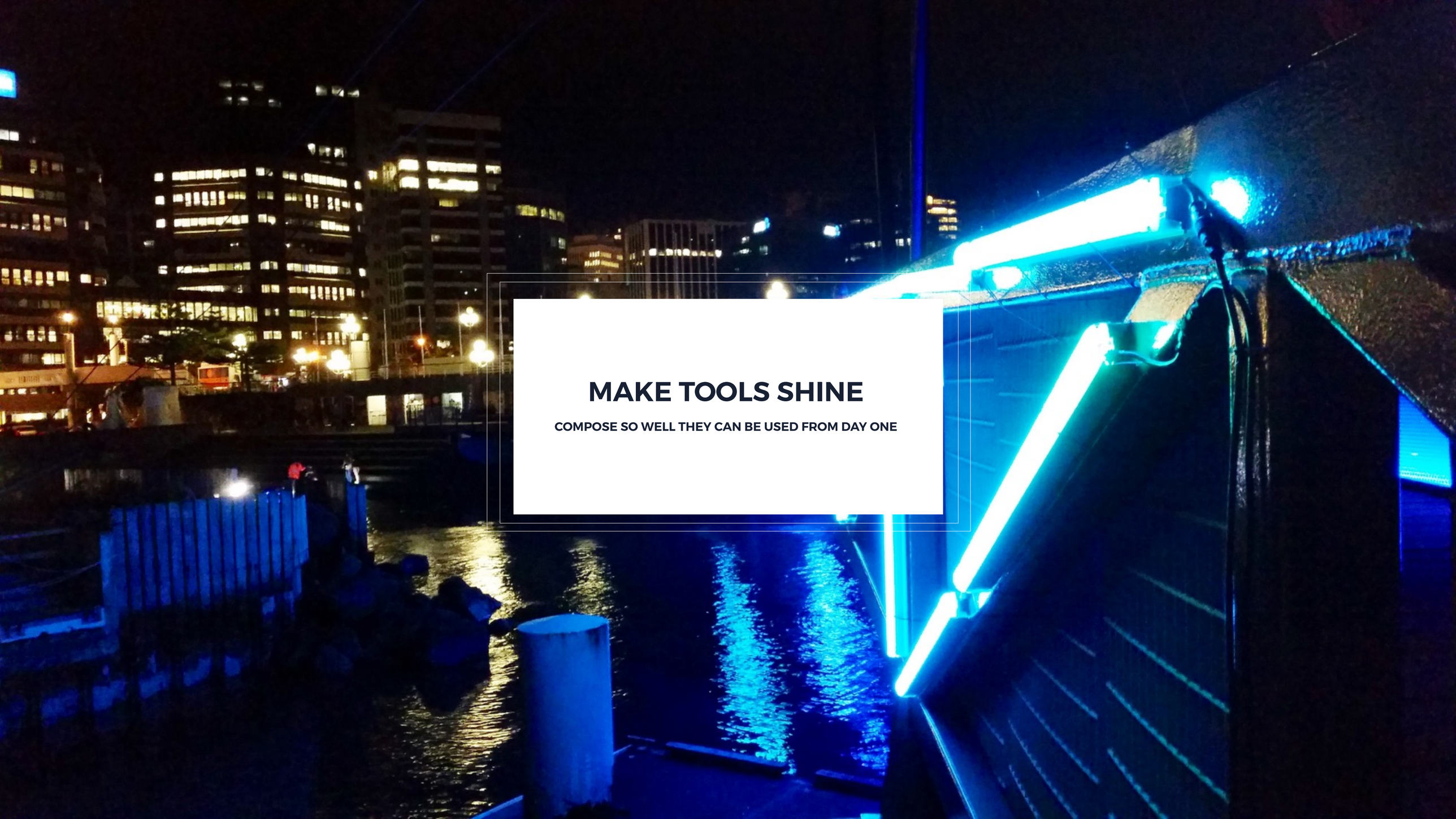
ideas to change how we do things



THERE IS NO QUICK FIX

SOLUTIONS TO PROBLEMS WE HAVE SHOULD

BE PUBLISHED AND REUSABLE



MAKE TOOLS SHINE

COMPOSE SO WELL THEY CAN BE USED FROM DAY ONE

MAKE YOUR TOOLS SHINE

HAVE SOME EMPATHY TOWARDS YOUR USERS

IT SHOULD BE FUN

LEARNING SHOULDN'T BE AN INVESTMENT

BUT A NATURAL PROGRESSION

let's try a thing

Image Attribution

01 **Into the Dragon's Lair**

Tobias Lindman // CC BY 2.0
flickr.com/photos/cowb0y2000/6243008944

02 **NASA N3-X hybrid wing aircraft**

NASA // Public Domain
wikimedia.org/wiki/File:NASA_N3-X_hybrid_wing_aircraft.jpg

03 **Instruction in Propeller Swinging**

Charles William Jefferys // Public Domain
wikimedia.org/wiki/File:Charles_William_Jefferys_-_Instruction_in_Propeller_Swinging.jpg

05 **Swindon Magic Roundabout**

Google Maps
https://goo.gl/RLDZfL

06 **H-4 Hercules "Spruce Goose"**

Federal Aviation Administration // Public Domain
wikipedia.org/wiki/File:H-4_Hercules_2.jpg

07 **A Midnight Modern Conversation**

William Hogarth, circa 1732 // Public Domain

10 **Irish spinner and spinning wheel**

Detroit Publishing Co. // Public Domain
commons.wikimedia.org/wiki/File:Elderlyspinnera.jpg

11 **Escape**

James Lee // CC BY 2.0
flickr.com/photos/jronaldlee/4579611880

Image Attribution

12 **Garrett Smith photo**

"[…] about my avatar […], no problem of course and please no attribution needed :)"

15 **Jayco 1006 Pop-up Camper Trailer**

Korey99 // CC BY-SA 3.0
wikimedia.org/wiki/File:JaycoPopupCamper-2009-11-17.jpg

16 **Hotell Anno 1647**

Some website about the hotel I forgot to note down :(

18 **Ecce Homo**

Elías García Martínez, circa 1930 // Public Domain

21 **Sunny Sand Desert Hiking**

CC0 License
pexels.com/photo/sunny-sand-desert-hiking-3853/

22 **Experiment with the air pump**

Joseph Wright of Derby, circa 1768 // Public Domain

— **Others**

My own.