

TRINITY COLLEGE SENIOR SEMINAR TALK

# TAMING COMPLEXITY IN SOFTWARE

**Real-life Stories and Lessons**

HYUNSU CHO, CLASS OF 2015

# ABOUT ME

2015

Graduate from  
**Trinity College**  
Enter Ph.D. @  
**University of Washington**

2017

Summer internship  
@ **AWS**  
Start managing  
**open source**  
**project (XGBoost)**

2018

Receive M.S. from  
Univ. of Wash.  
Join **AWS** as  
Applied Scientist

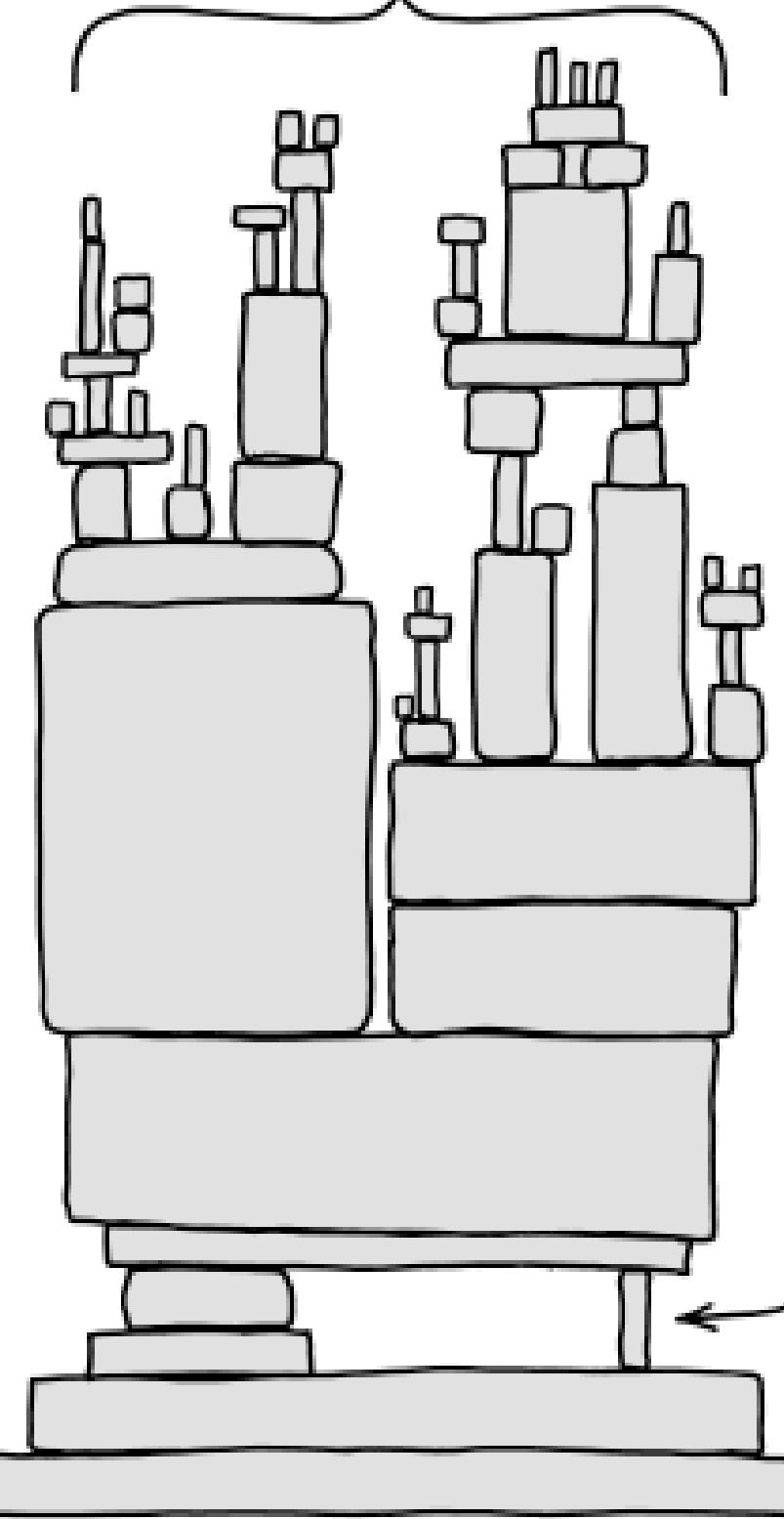


2020

Join **NVIDIA** as  
Senior Systems  
Software  
Engineer



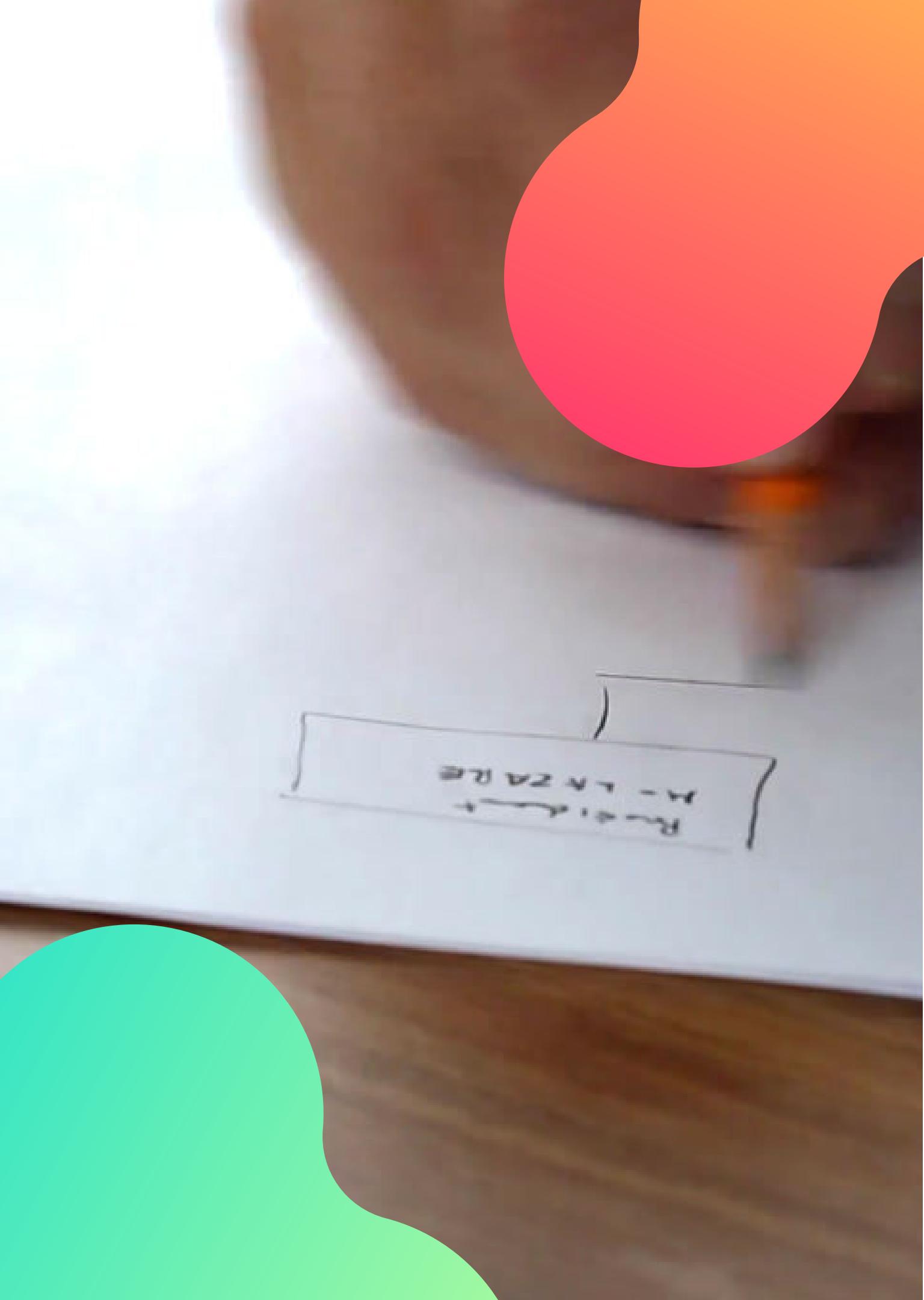
ALL MODERN DIGITAL  
INFRASTRUCTURE



<https://xkcd.com/2347/> "Dependency"

# TODAY'S TOPIC

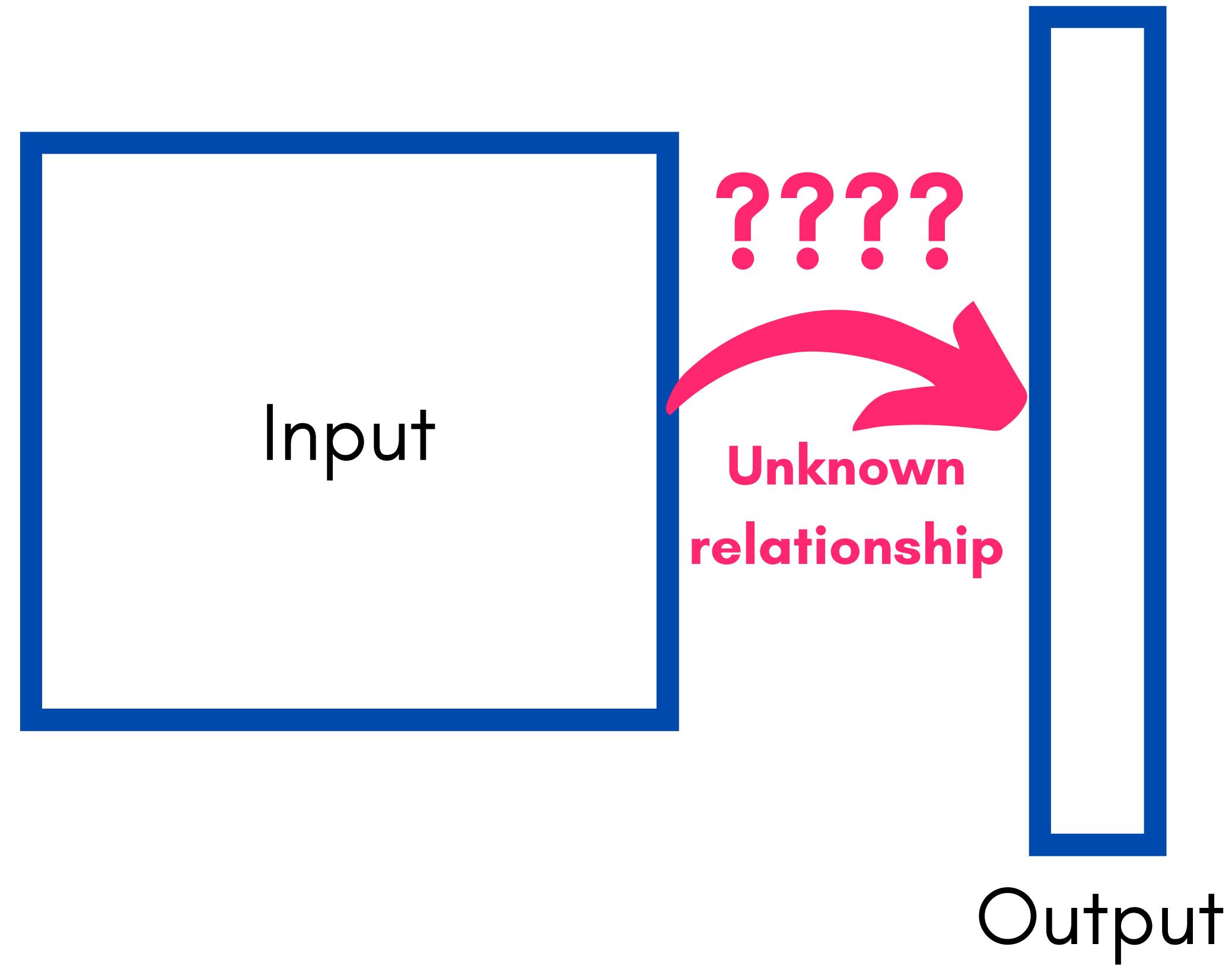
How to deal with software complexity

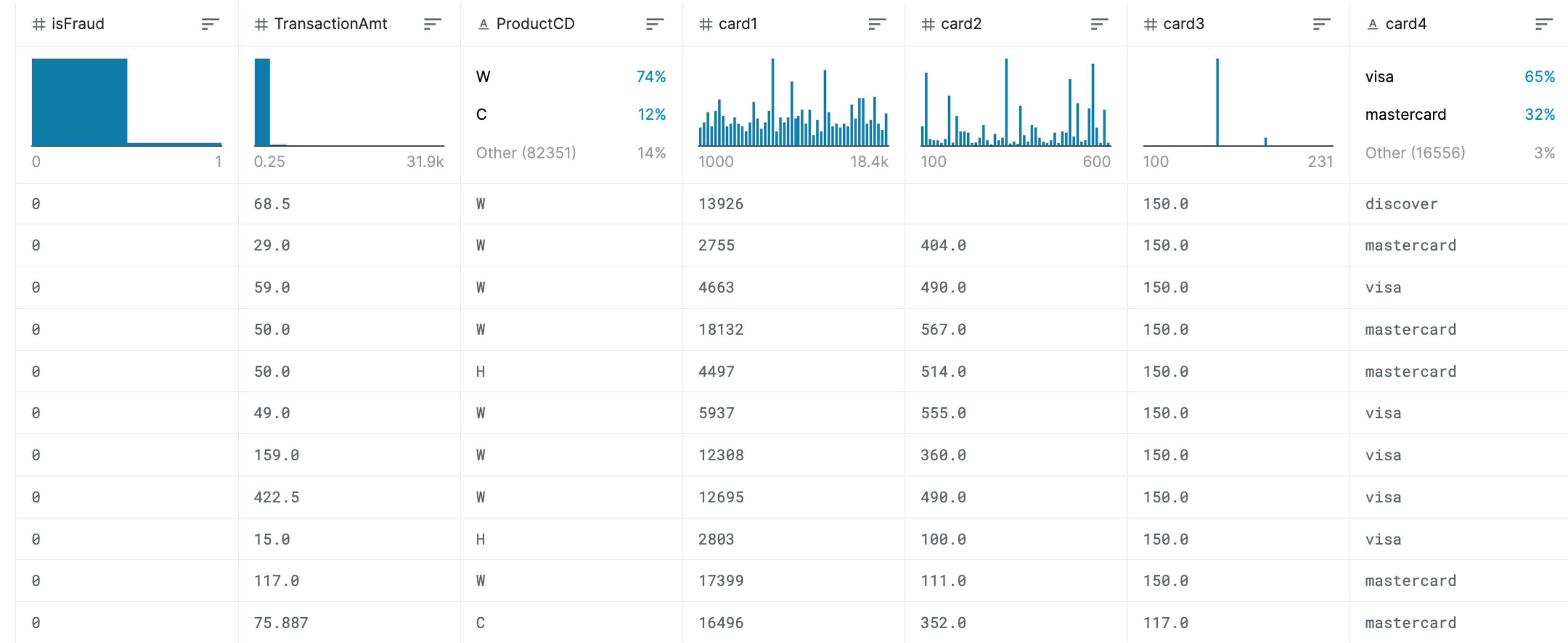


# VERY SHORT INTRO TO “AI”

# Super hot topic

## What I do at my day job





Example: credit card transaction (some fraudulent)

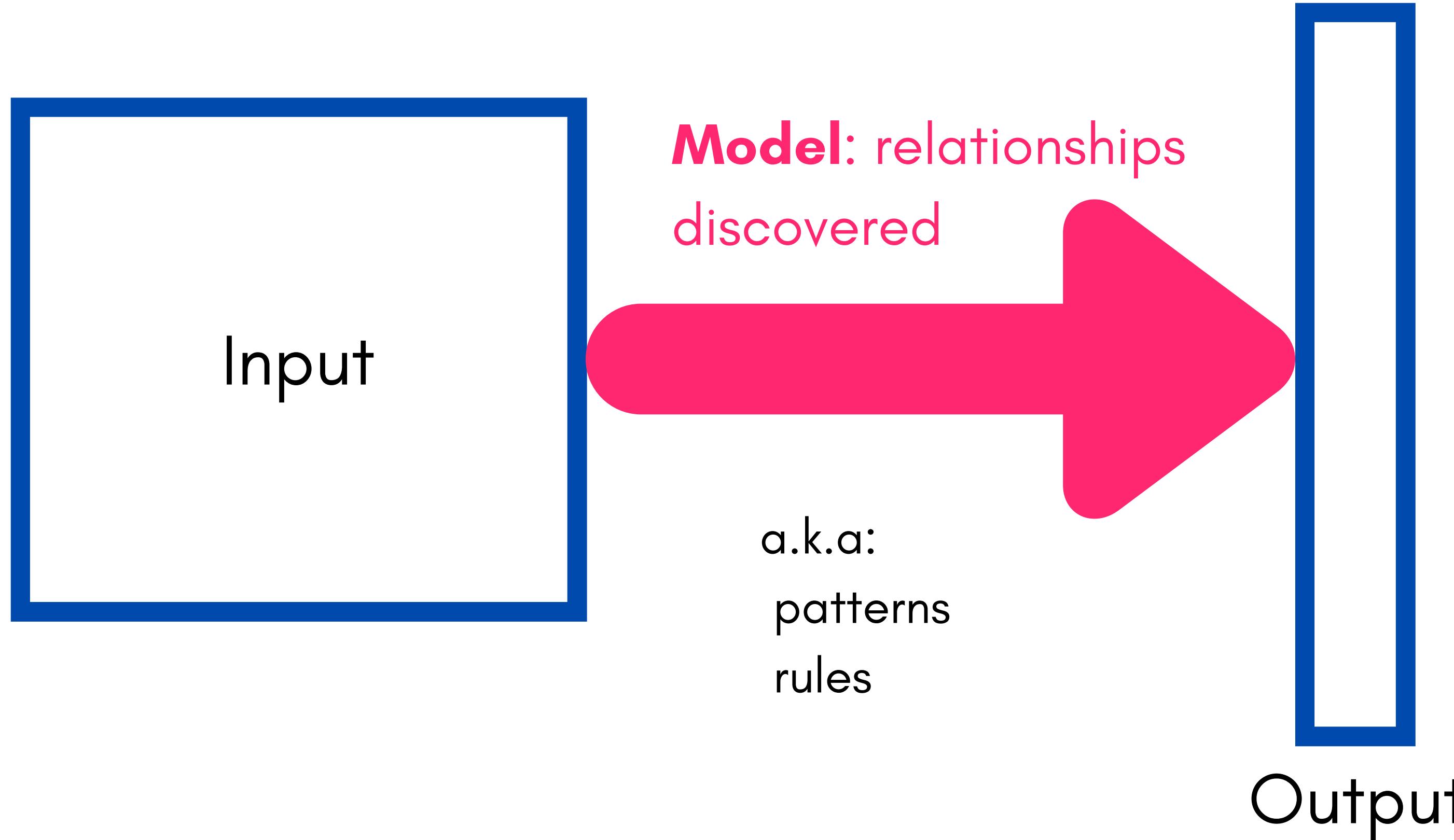
# Output: fraud? (Y/N)

# Input: attributes of transactions

# isFraud	# TransactionAmt	▲ ProductCD	# card1	# card2	# card3	▲ card4
0	0.25	W	74%	1000	18.4k	visa 65%
0	68.5	C	12%	100	600	mastercard 32%
0	29.0	Other (82351)	14%	100	600	Other (16556) 3%
0	59.0	W	13926	150.0	discover	
0	50.0	W	2755	404.0	150.0	mastercard
0	50.0	W	4663	490.0	150.0	visa
0	50.0	W	18132	567.0	150.0	mastercard
0	49.0	H	4497	514.0	150.0	mastercard
0	159.0	W	5937	555.0	150.0	visa
0	422.5	W	12308	360.0	150.0	visa
0	15.0	W	12695	490.0	150.0	visa
0	117.0	H	2803	100.0	150.0	visa
0	75.887	C	17399	111.0	150.0	mastercard
0			16496	352.0	117.0	mastercard

Example: credit card transaction (some fraudulent)

# AI **discovers relationship** between input and output



# THE AI WORKFLOW

## 1 Collect data

Need high-quality data, in great quantity

## 3 Test the model

Test the model with similar data and assess accuracy. If bad, go back to Step 2.

## 2 Train a model

Model **discovers all interesting patterns** from the data

## 4 Deploy to production

Search engine, credit card app server, smart-speakers etc.

# THE AI WORKFLOW

**Deployment is HARD!**



4

**Deploy to production**

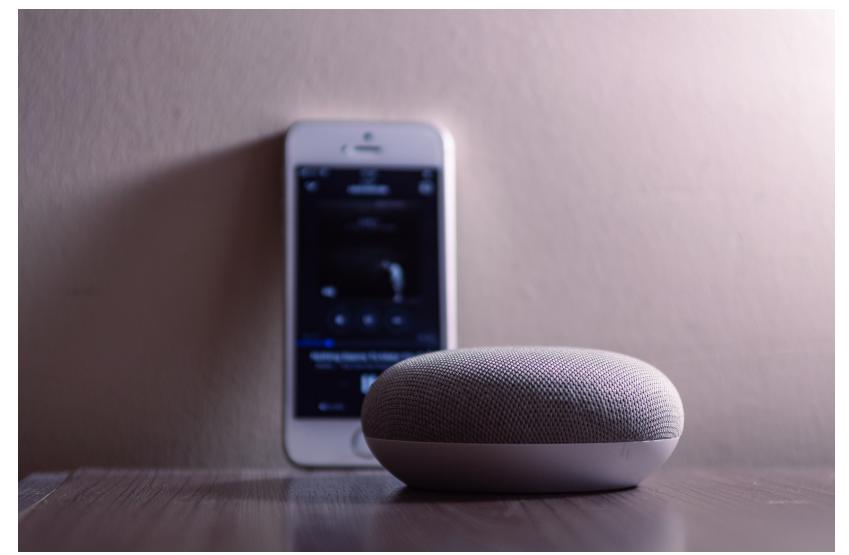
Search engine, credit card app  
server, smart-speakers etc.



AI model



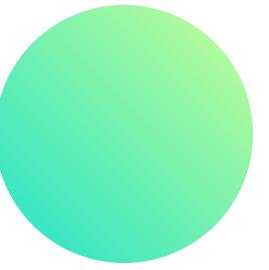
Web



Edge device

# **TWO LIFE STORIES**

My early mistakes, and lessons learned





# STORY #1

---

## **My internship project goes into production**

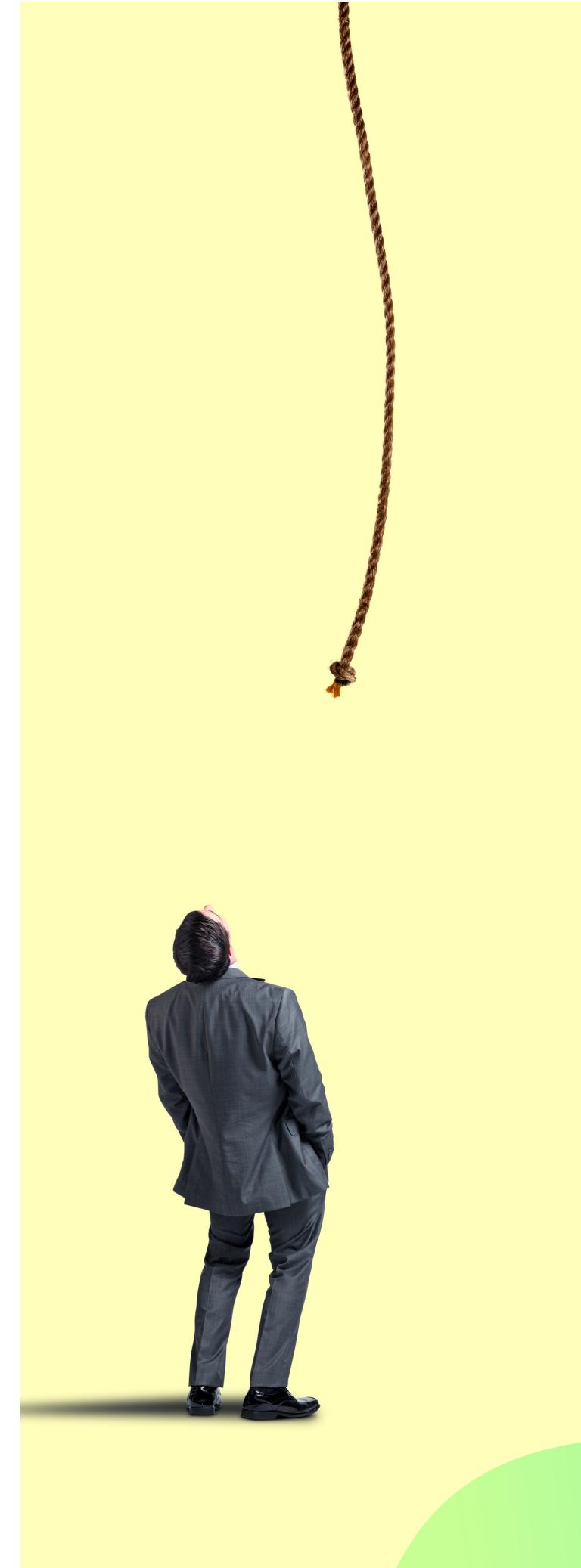
How a "prototype" made it to  
production and hell broke loose

# JAN 2017

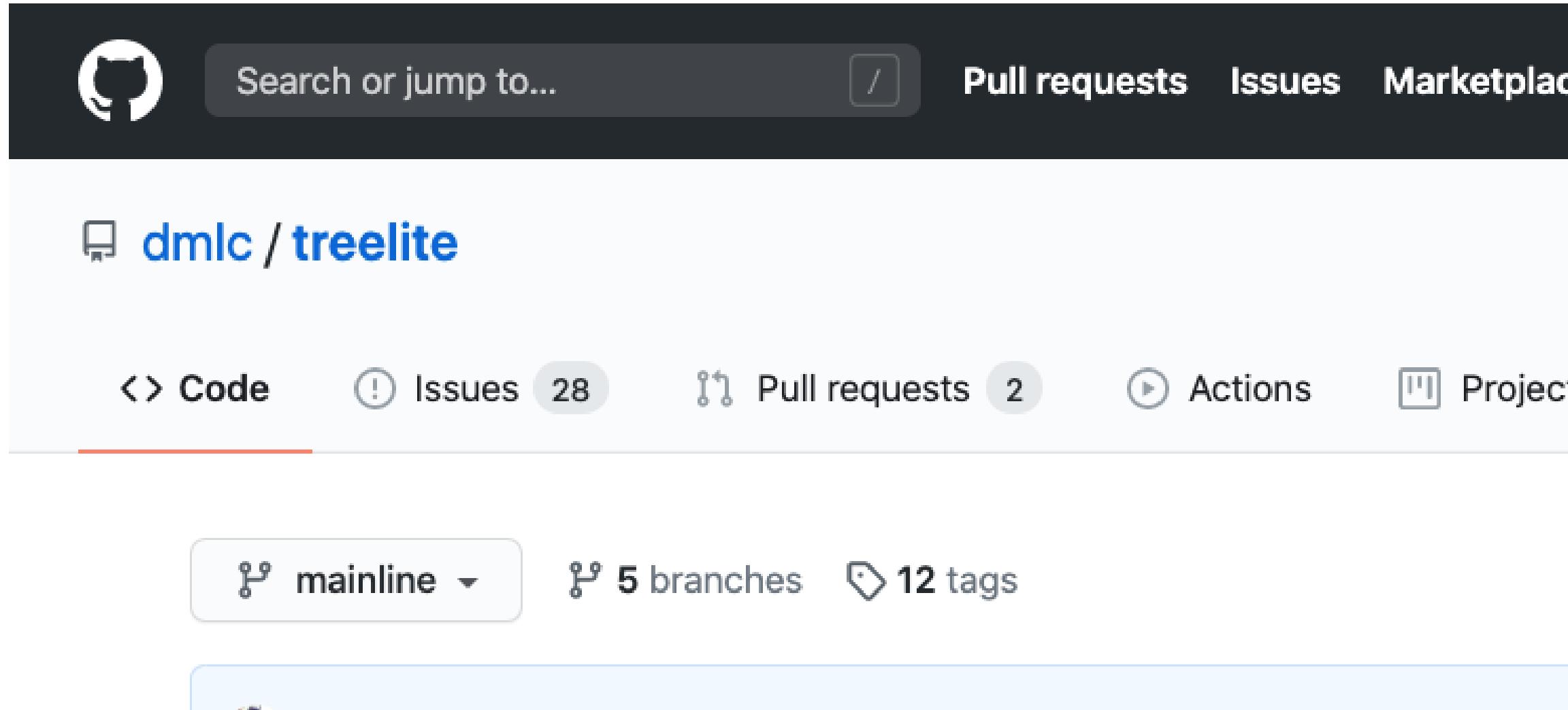


*How my Ph.D. ended*

# JULY 2017



# JULY-DEC 2017



Got me a full-time job



Brand new AI tool  
30k lines of code from  
scratch

# Happy Ever After?

...Not so fast!

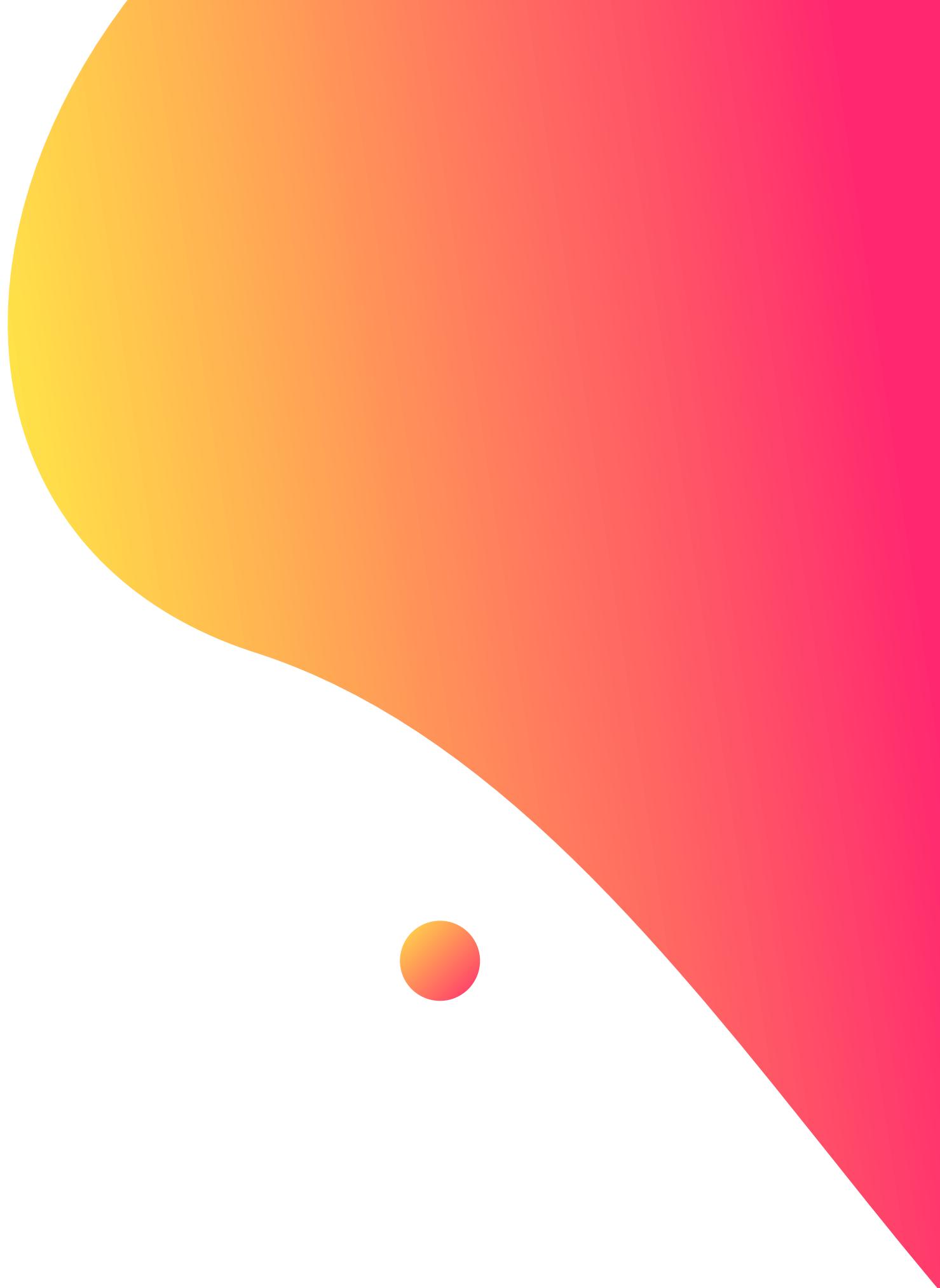
# SUMMER 2018 @ AWS: “AI AS SERVICE”

---

My intern project became part  
of a new AWS service!

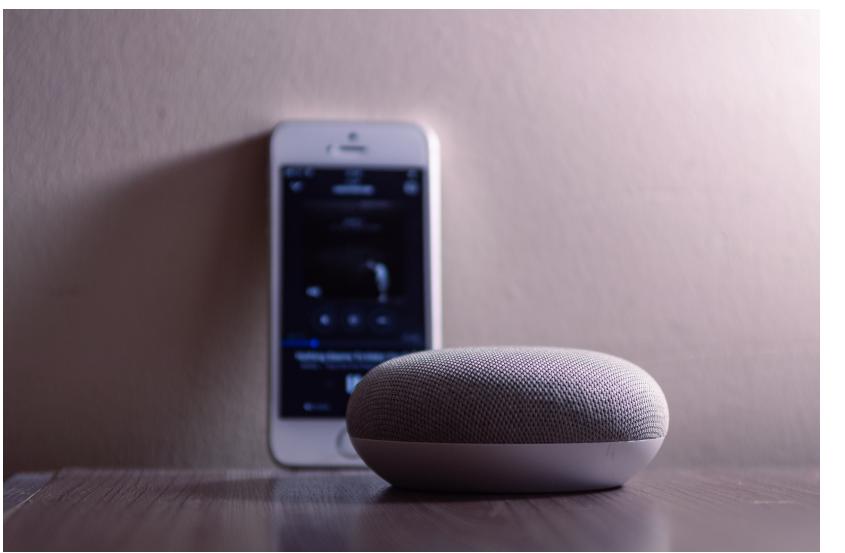
**Good:** Got promoted

**Bad:** Saw my prototype crash in  
production like a house of cards





AI model



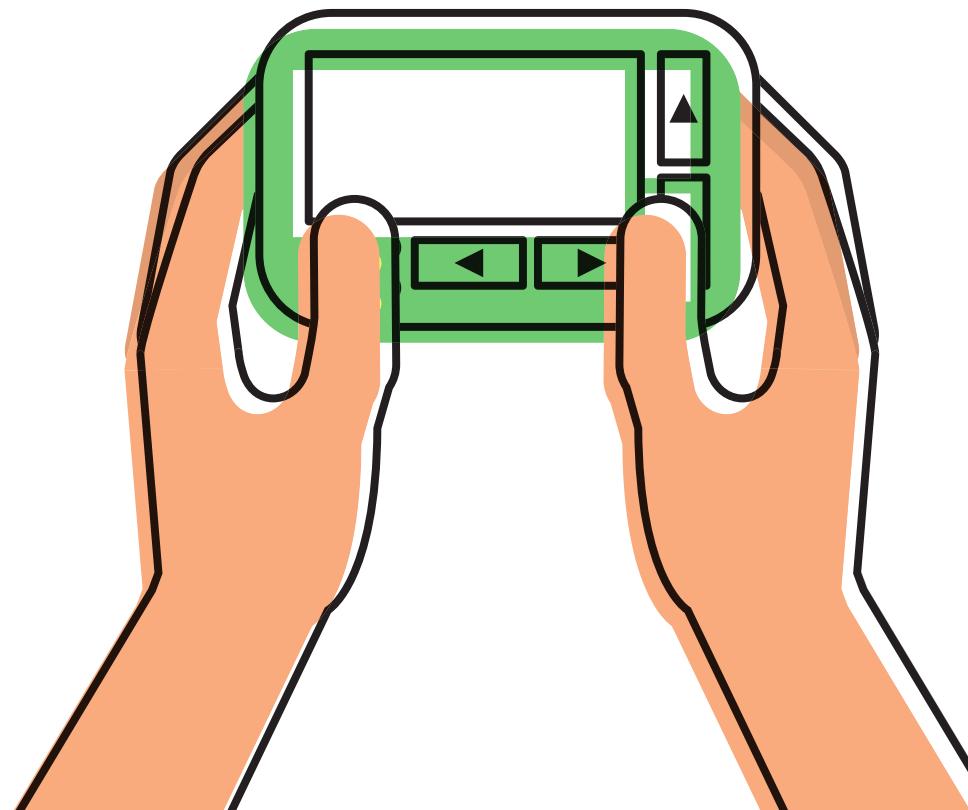
Edge device

# THE ON-CALL EXPERIENCE

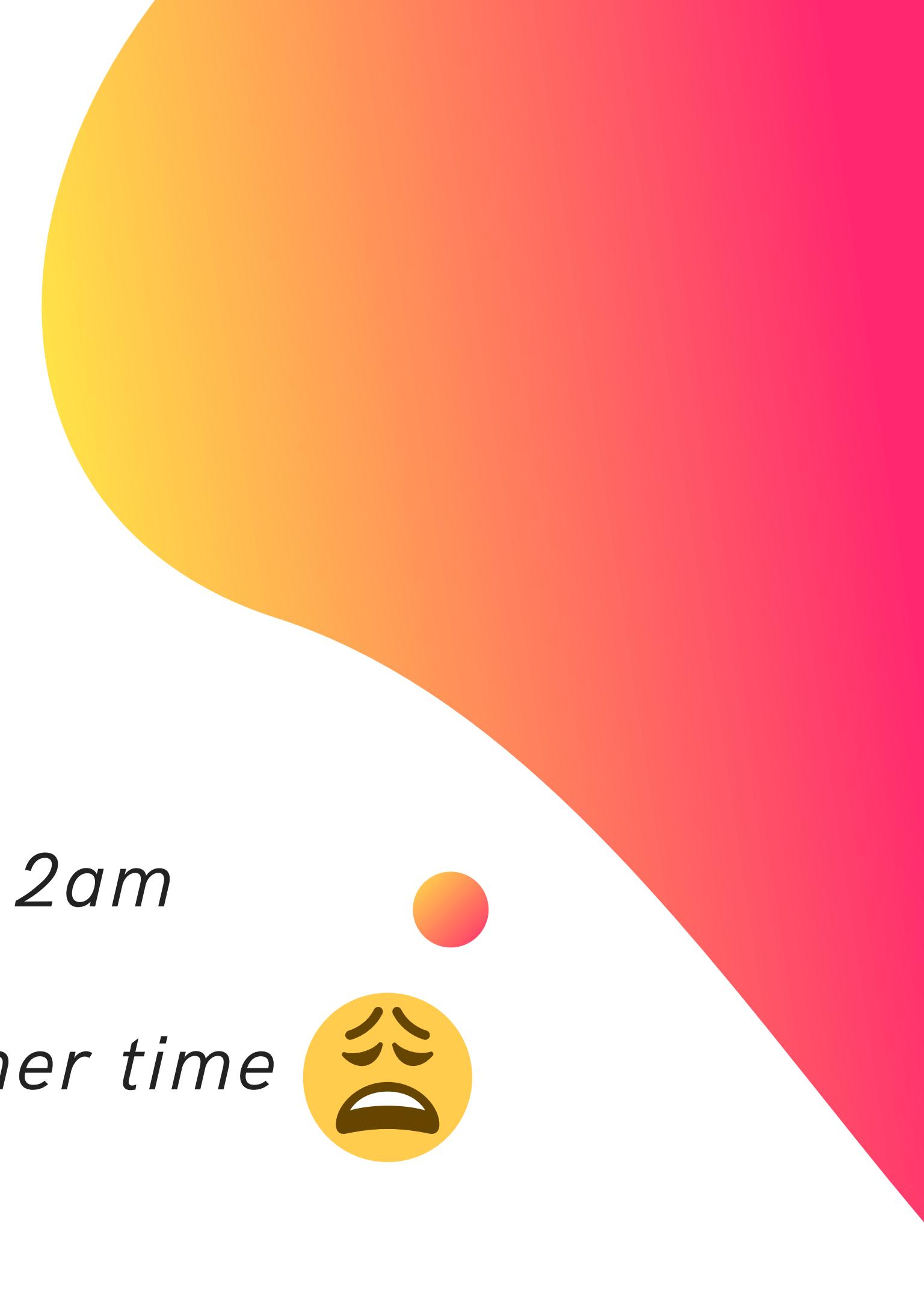
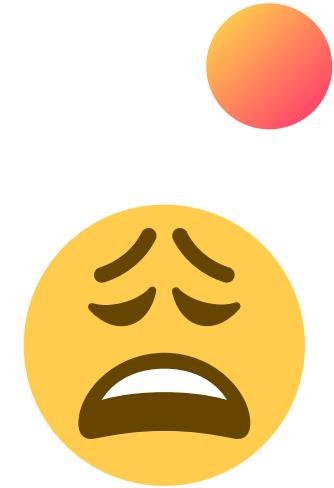
---

Service must be up 24/7

Everyone in the team had  
on-call rotation



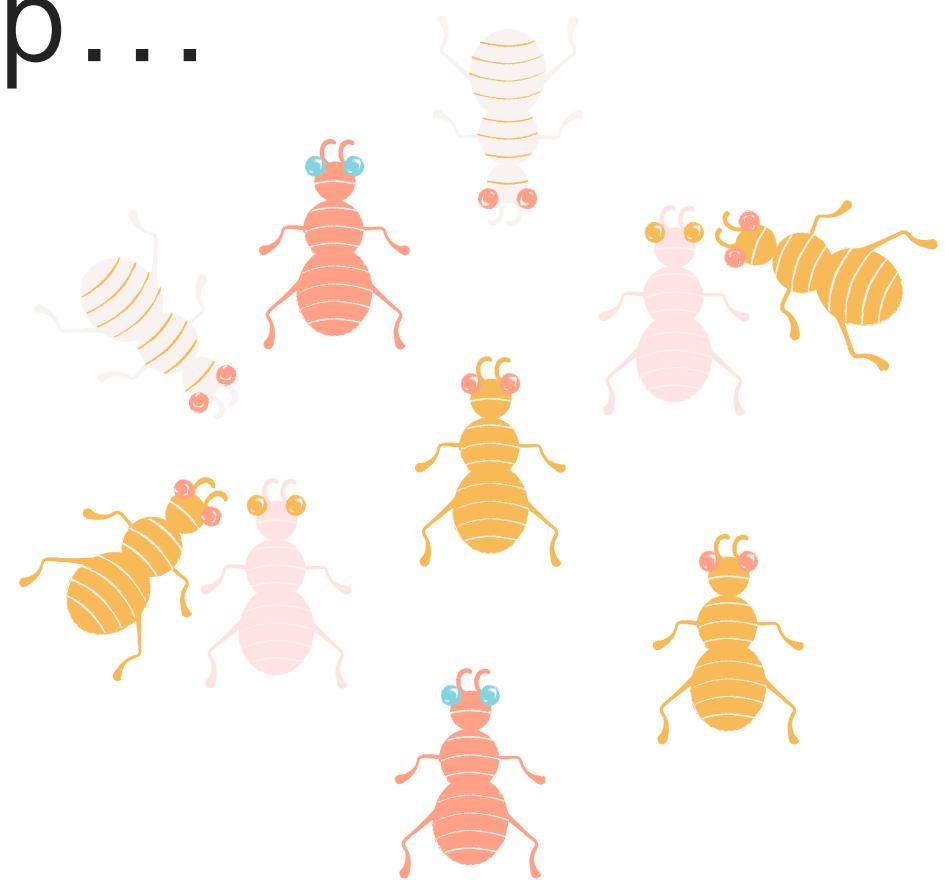
*Pager goes off at 2am  
Realize your code  
crashed yet another time*



# BUGS AFTER BUGS

---

My code revealed lots of bugs  
after going into production.  
Spent several months fixing  
bugs. Tickets piled up...



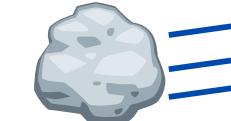
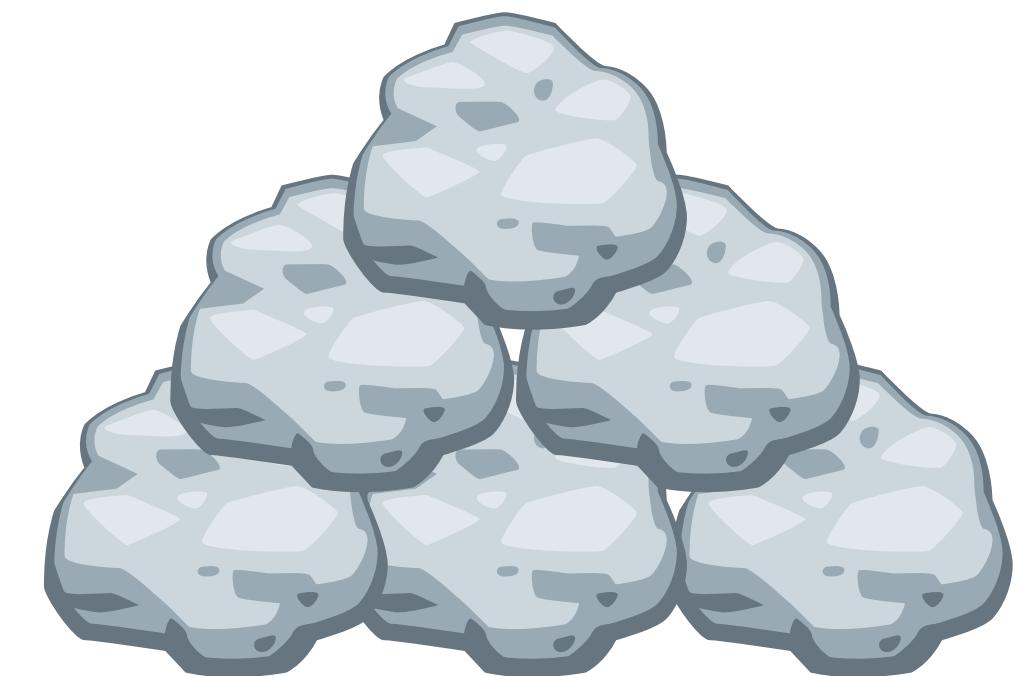
# BUGS AFTER BUGS

My code revealed lots of bugs  
after going into production.  
Spent several months fixing  
bugs. Tickets piled up...



# WHAT WENT WRONG??

I had thrown together bunch of code to get a **prototype** working quickly. Multiple parts were tightly glued together



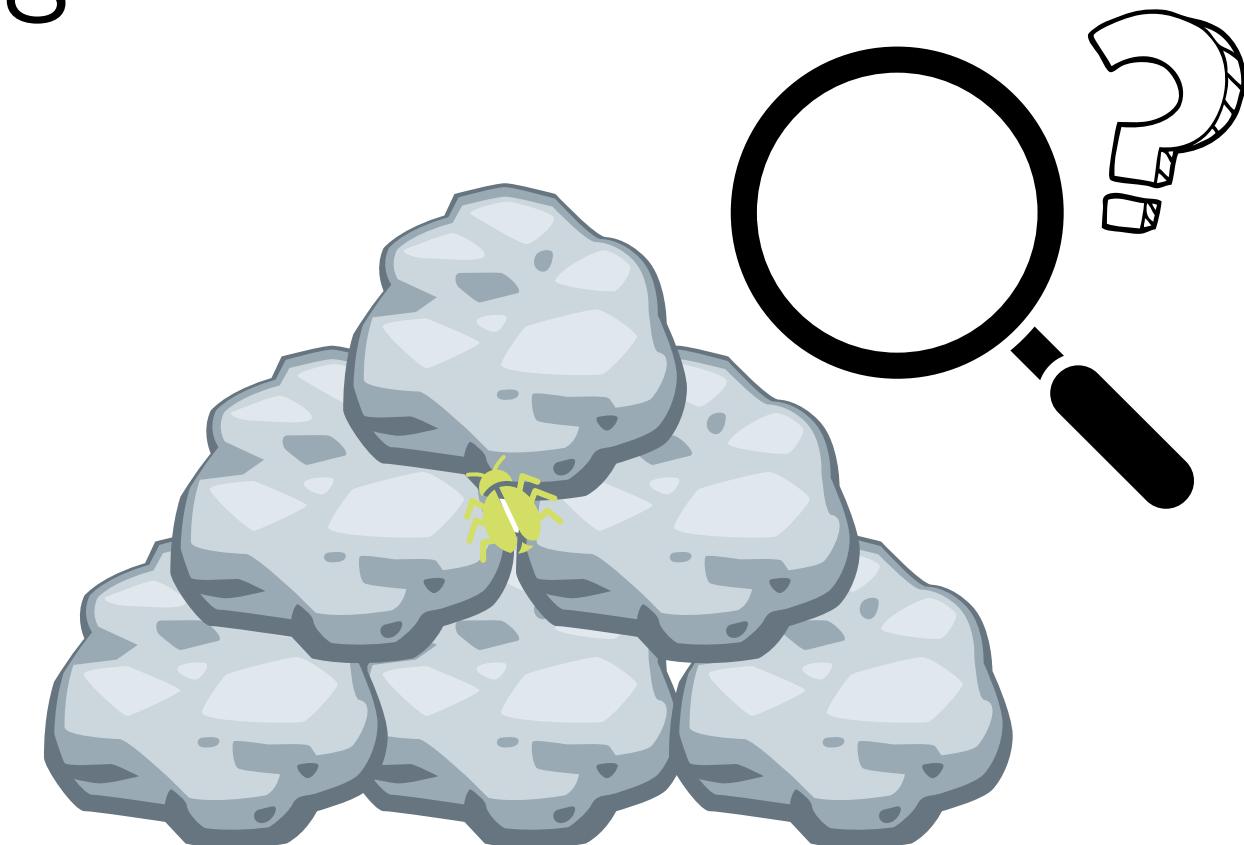
*Gonna throw together  
something quick! Hope  
it works!*



*Me (intern)*

# WHAT WENT WRONG??

I had thrown together bunch of code to get a prototype working quickly. Multiple parts were tightly glued together



Hard to find bug  
Hard to know  
which part crashed

# WHAT WENT WRONG??

I had thrown together bunch of code to get a prototype working quickly. Multiple parts were tightly glued together

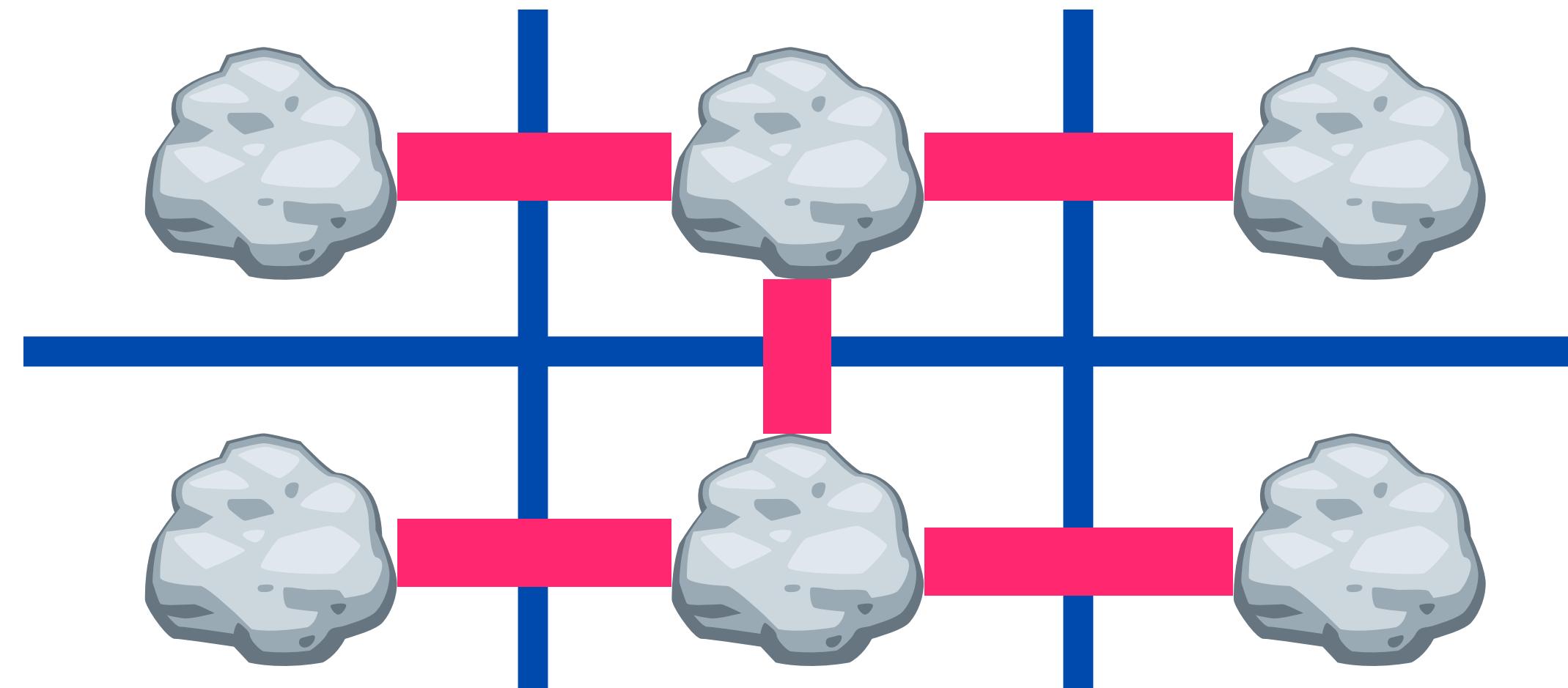


Hard to add a new feature, since you have to change lots of other places ("big blast radius")

# WHAT WENT WRONG??

If I wrote a modular, cleanly separated code,  
I would have saved lots of pain later.

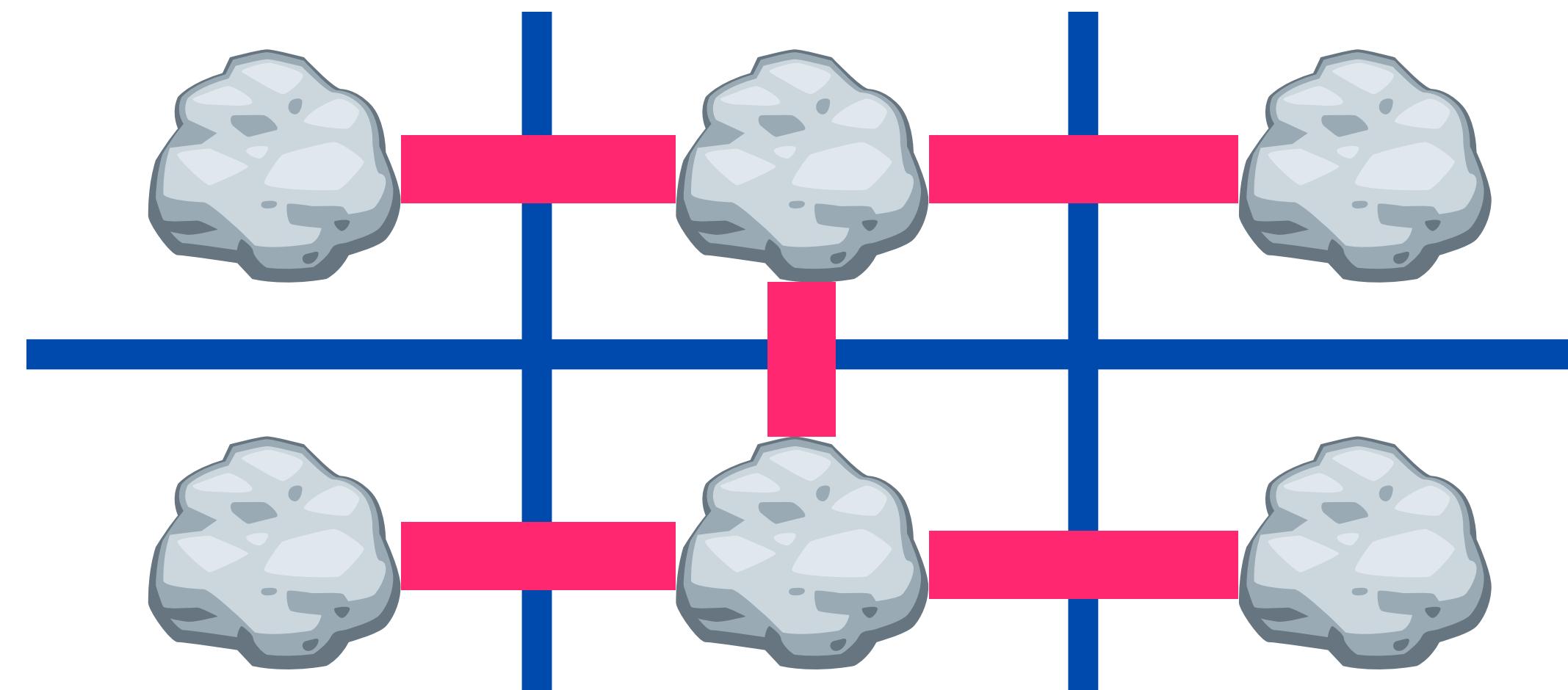
Each module does  
only one thing  
(separation of  
concerns)



# WHAT WENT WRONG??

If I wrote a modular, cleanly separated code,  
I would have saved lots of pain later.

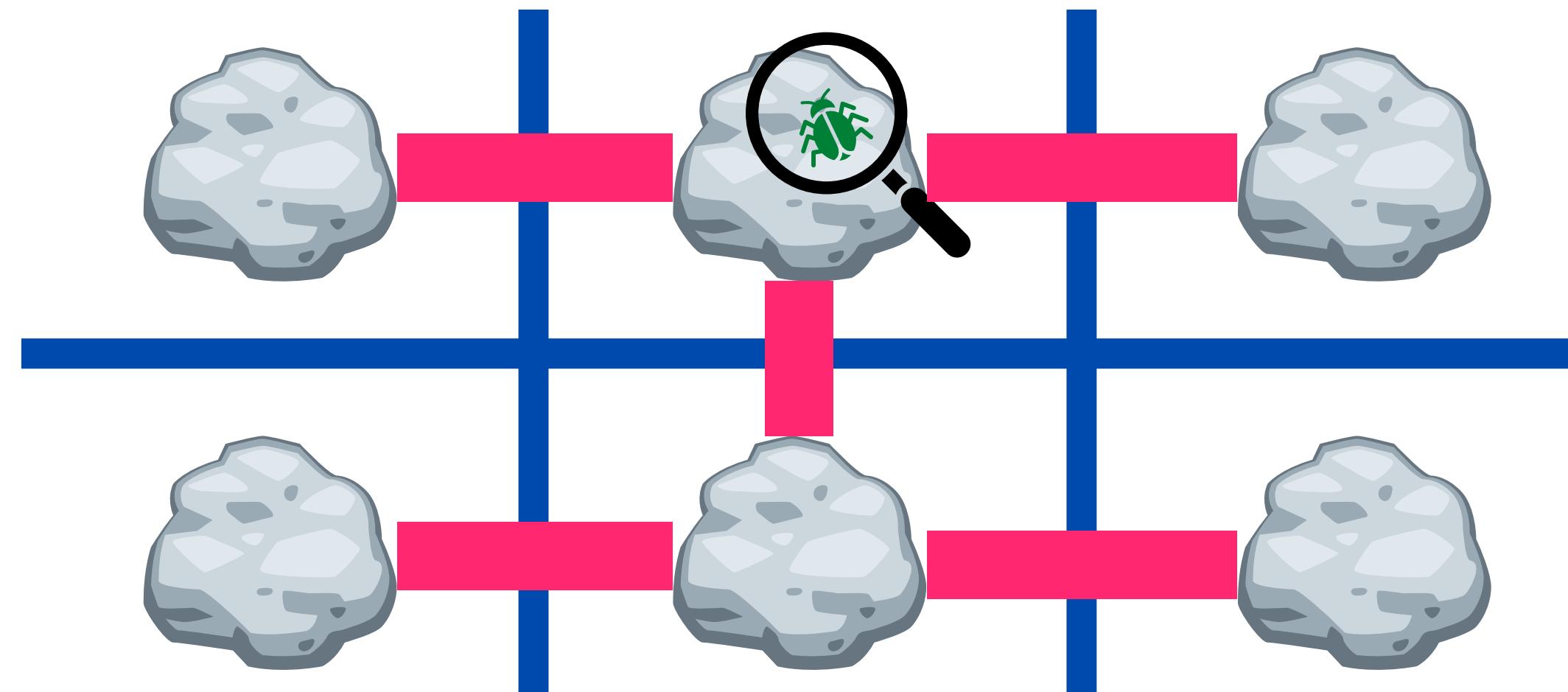
Modules  
communicate with  
each other only in a  
few places  
(decoupling)



# WHAT WENT WRONG??

If I wrote a modular, cleanly separated code,  
I would have saved lots of pain later.

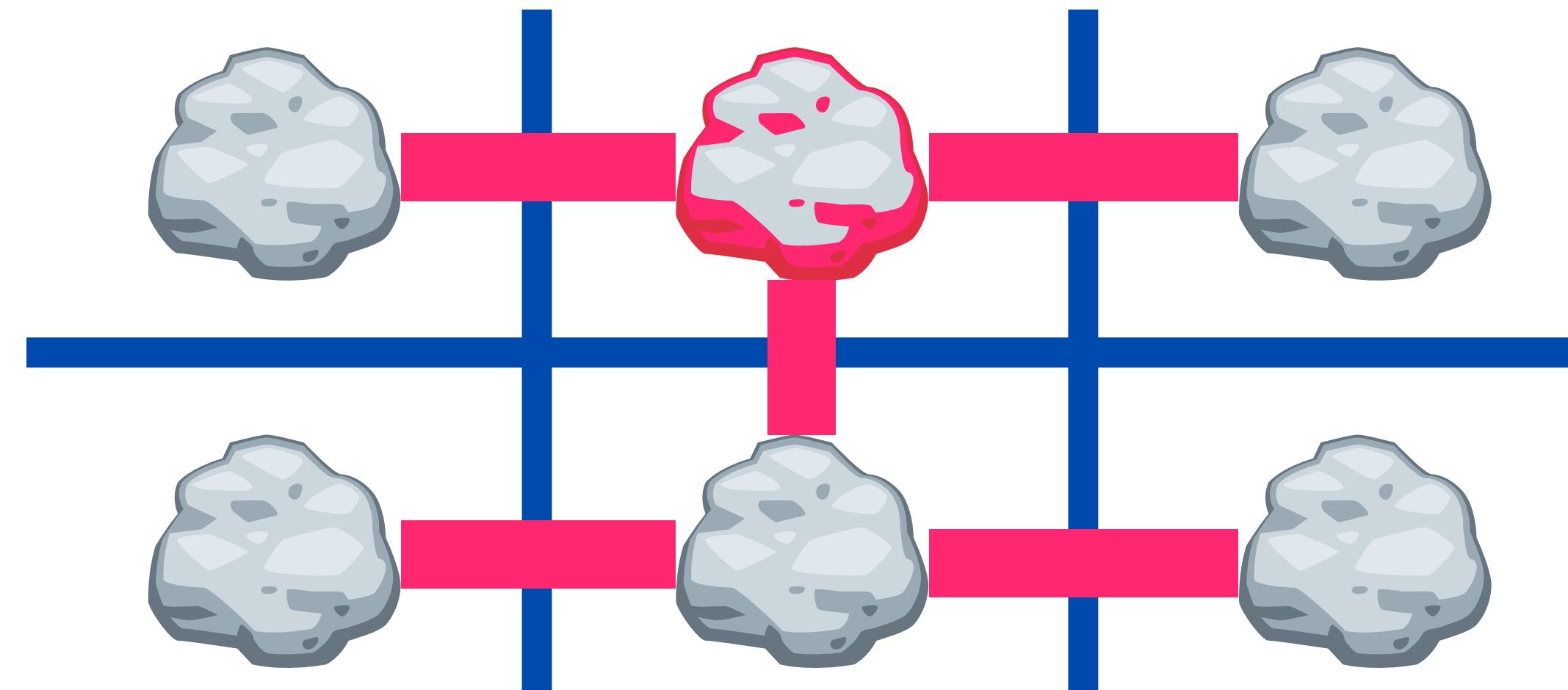
Easy to find which  
module crashed  
  
Can **test** each  
module one by one



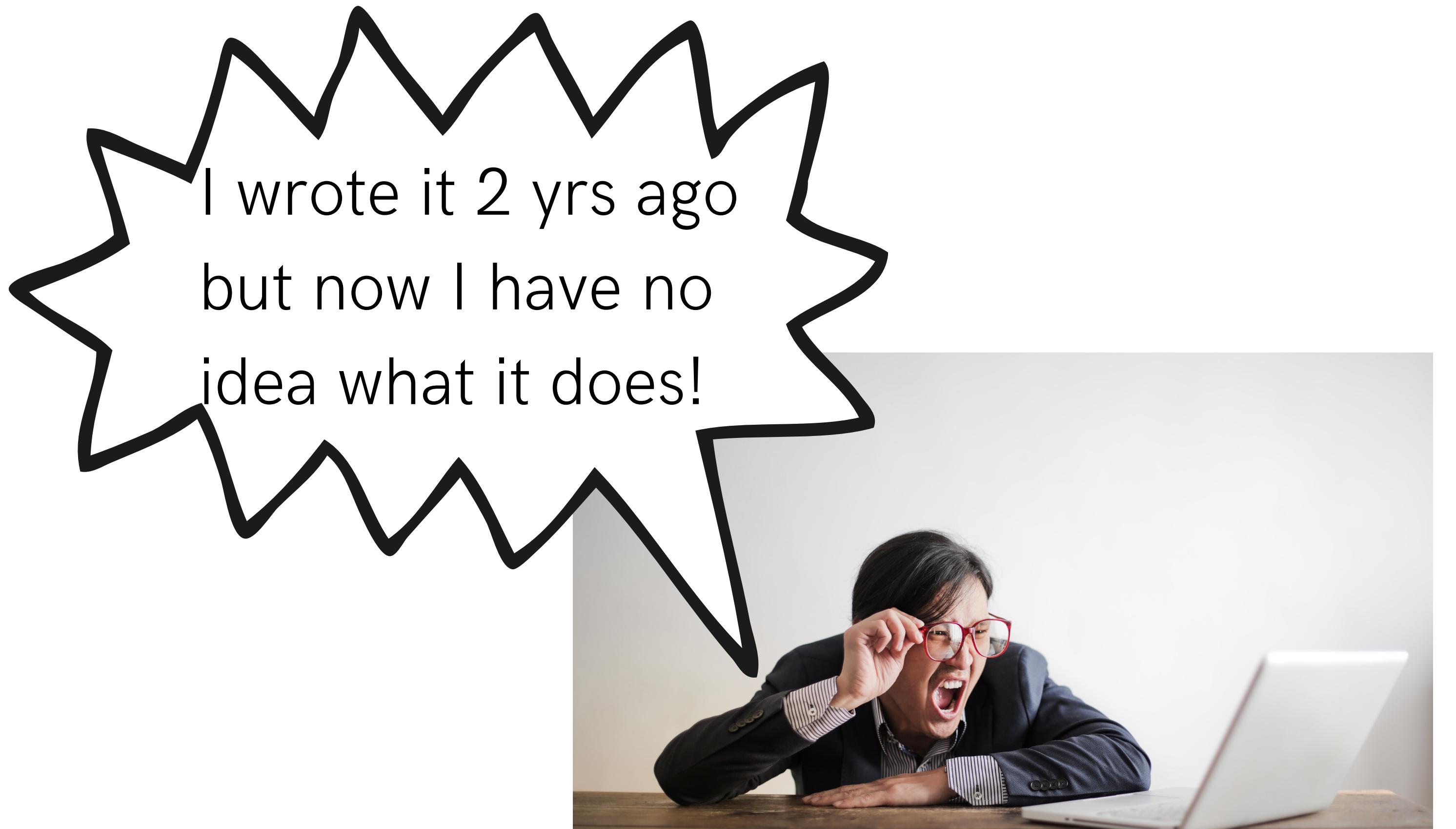
# WHAT WENT WRONG??

If I wrote a modular, cleanly separated code,  
I would have saved lots of pain later.

Easy to swap out  
one module and  
put in another,  
while keeping other  
modules

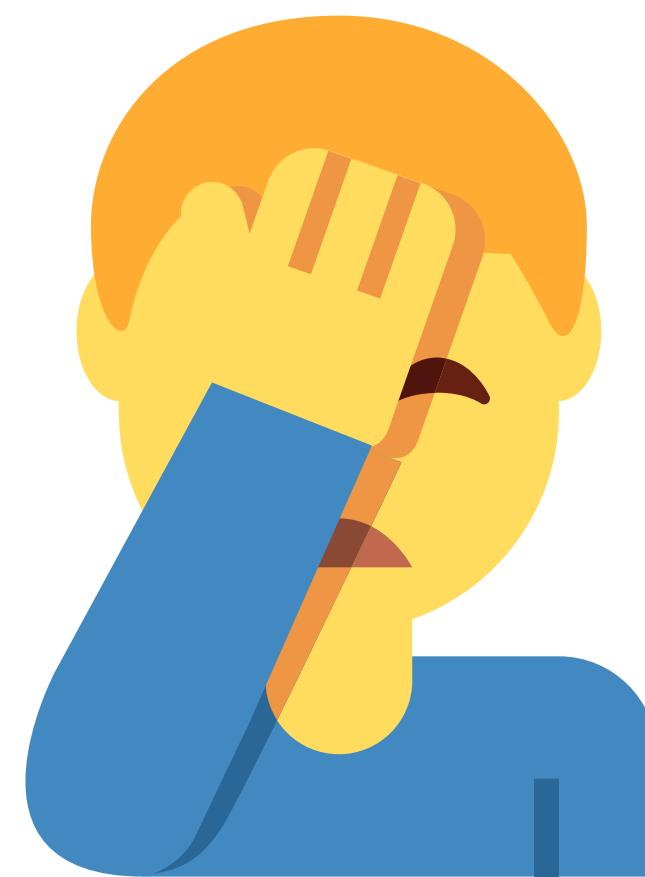


# WHAT WENT WRONG??



# AFTERMATH: MEGA REWRITE (AUGUST 2020)

+4,180 -3,406



*Me, this year*

# STORY #2

# The importance of “no”

# How I admitted a questionable pull request and ended up hurting the project



# THE OPEN SOURCE MOVEMENT

---

Wikipedia for Code

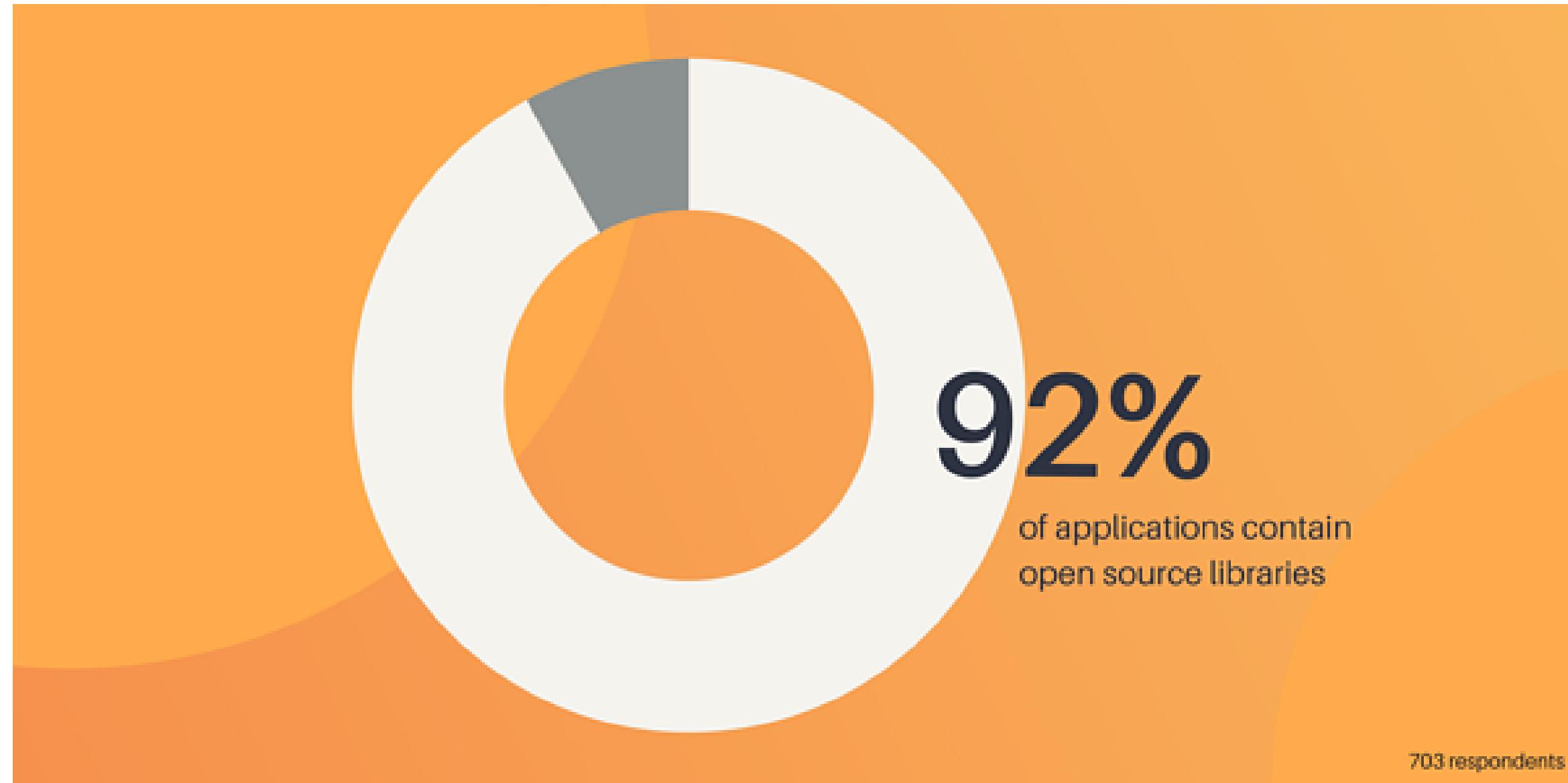
Code is open; anyone can contribute

Massive community effort

Big impact

Lots of great software developed this way





Tidelift. "Open source is everywhere"  
[https://blog.tidelift.com/open-source-is-everywhere-survey-  
results-part-1](https://blog.tidelift.com/open-source-is-everywhere-survey-results-part-1)

# GitHub

platform for  
hosting open  
source projects

The screenshot shows a GitHub pull request page for the `xgboost` repository. The title of the pull request is `[CI] Build a Python wheel for aarch64 platform #6253`. The status of the pull request is `Merged`, indicating it has been merged into the `master` branch from the `build_aarch64_wheel` branch yesterday. The page displays the code changes made in this pull request, specifically focusing on the `.travis.yml` and `Dockerfile.aarch64` files.

**.travis.yml**

```
5 @@ -15,6 +15,11 @@ jobs:  
15   - os: linux  
16     arch: arm64  
17     env: TASK=python_sdist_test  
  
18   - os: osx  
19     arch: amd64  
20     osx_image: xcode10.1  
  
...
```

**Dockerfile.aarch64**

```
37 @@ -0,0 +1,37 @@  
  
1 FROM quay.io/pypa/manylinux2014_aarch64  
2 +  
3 SHELL ["/bin/bash", "-c"] # Use Bash as shell  
4 +  
5 # Install all basic requirements  
6 RUN \  
7 + yum update -y && \  
8 + yum install -y tar unzip wget xz git centos-release-scl-rh yum-utils && \  
9 + yum-config-manager --enable centos-scl0-rh-testing && \  
10 + yum update -y && \  
11 + yum install -y devtoolset-7 && \  
12 + # Python  
13 + wget --no-verbose -O conda.sh https://github.com/conda-forge/miniforge/releases/download/4.8.2-  
1/Miniforge3-4.8.2-1-Linux-aarch64.sh && \  
14 + bash conda.sh -b -p /opt/miniforge-python  
15 +  
16 + ENV PATH=/opt/miniforge-python/bin:$PATH  
...
```

# GitHub

platform for  
hosting open  
source projects

Search or jump to... Pull requests Issues Marketplace Explore

dmlc/xgboost Sponsor Unwatch 984 Unstar 20k Fork 7.8k

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

## [CI] Build a Python wheel for aarch64 platform #6253

Merged hcho3 merged 21 commits into dmlc:master from hcho3:build\_aarch64\_wheel yesterday

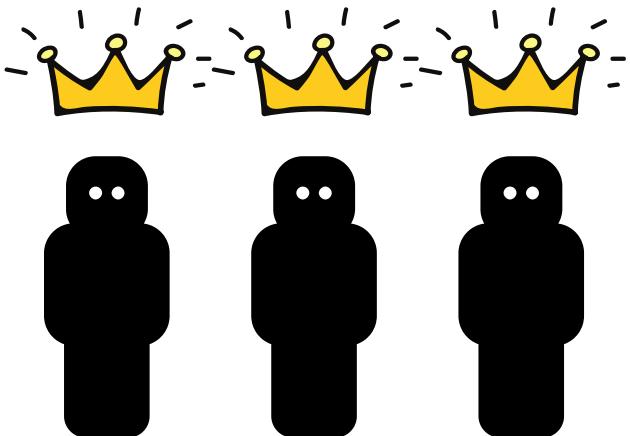
Conversation 10 Commits 21 Checks 24 Files changed 7

Changes from all commits File filter... Jump to... Review changes

# Proposed changes

```
@@ -15,6 +15,11 @@ jobs:
15   - os: linux
16     arch: arm64
17     env: TASK=python_sdist_test
18   - os: osx
19     arch: amd64
20     osx_image: xcode10.1
...
37 tests/ci_build/Dockerfile.aarch64
...
@@ -0,0 +1,37 @@
+ FROM quay.io/pypa/manylinux2014_aarch64
+
+ SHELL ["/bin/bash", "-c"] # Use Bash as shell
+
+ # Install all basic requirements
+ RUN \
+     yum update -y && \
+     yum install -y tar unzip wget xz git centos-release-scl-rh yum-utils && \
+     yum-config-manager --enable centos-scl0-rh-testing && \
+     yum update -y && \
+     yum install -y devtoolset-7 && \
+     # Python
+     wget --no-verbose -O conda.sh https://github.com/conda-forge/miniforge/releases/download/4.8.2-1/Miniforge3-4.8.2-1-Linux-aarch64.sh && \
+     bash conda.sh -b -p /opt/miniforge-python
+
+ ENV PATH=/opt/miniforge-python/bin:$PATH
```

## Approve or reject pull requests

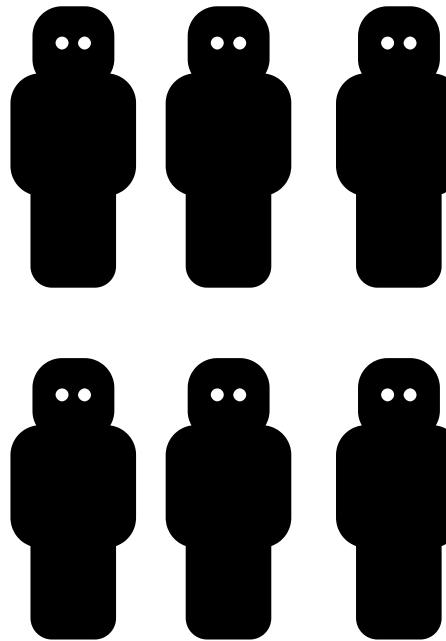


Maintainers

Vets contributions for  
the long-term benefit  
of the project



Pull request  
(proposed changes)



Contributors



≡ **dmlc / xgboost**

Sponsor

Watch

984

Star

20k

Fork

7.8k



Tianqi Chen  
Original creator

Currently Assistant Prof. @  
Carnegie Mellon Univ.

Popular AI package

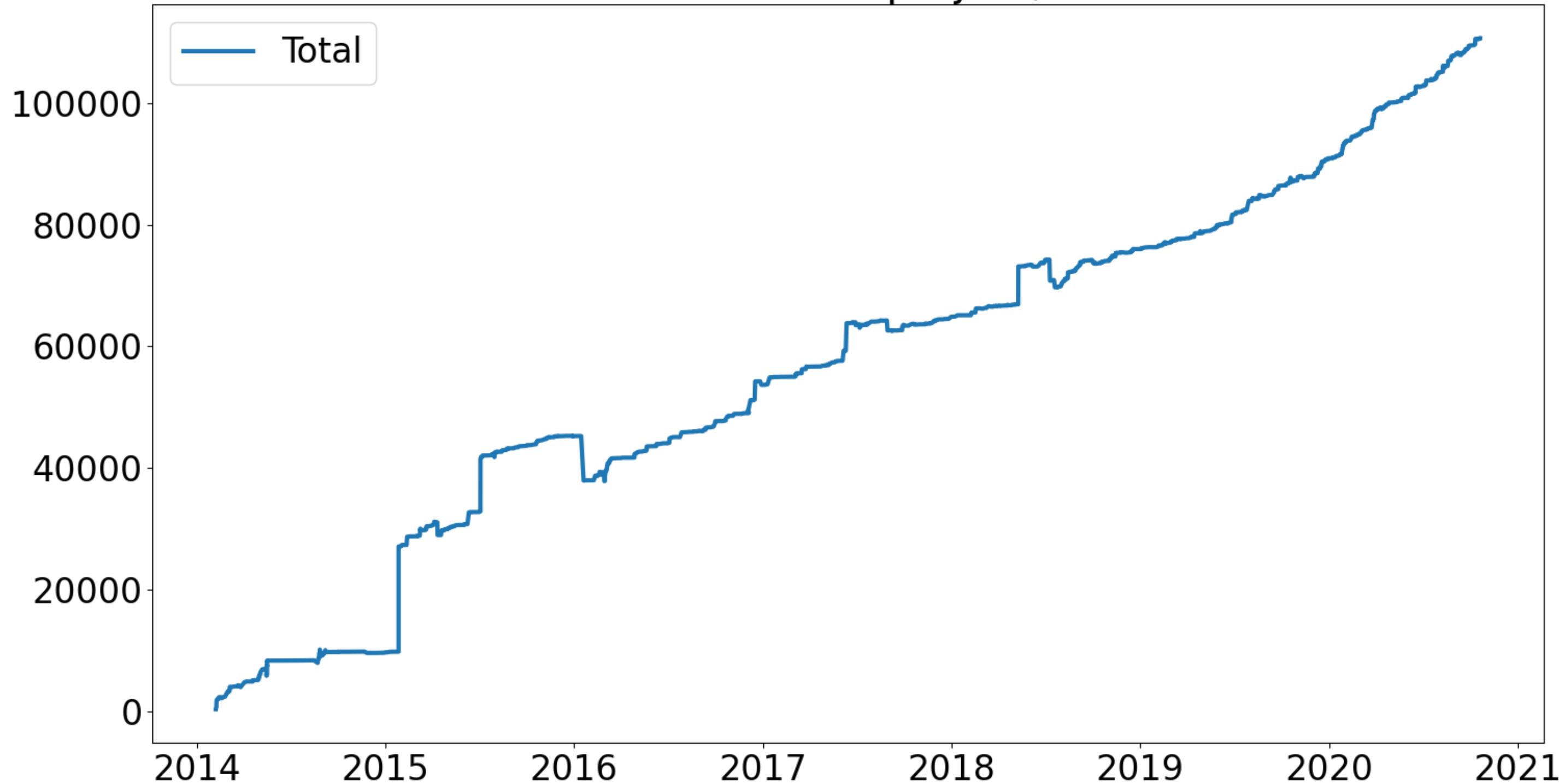
20k+ stars

100k downloads/day

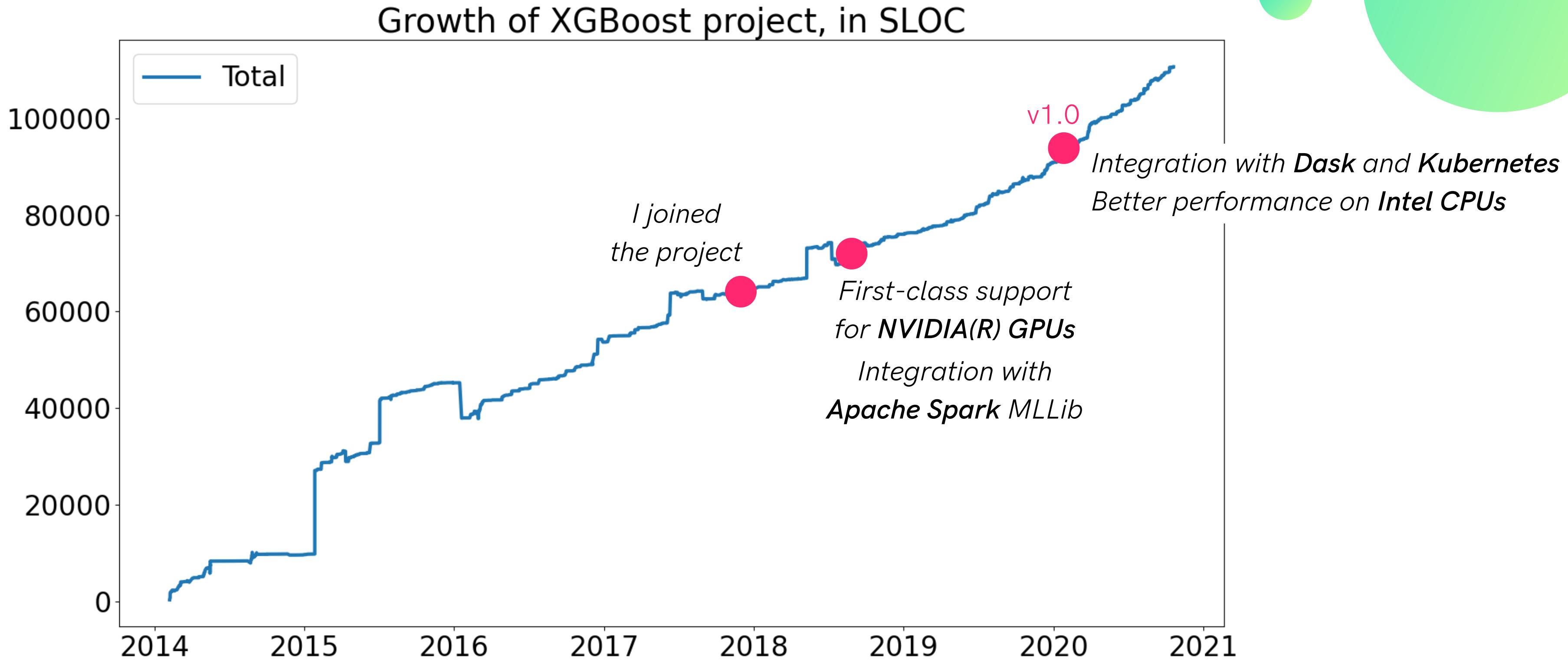
I was appointed as a  
maintainer since **Nov 2017**

# GROWING PAINS

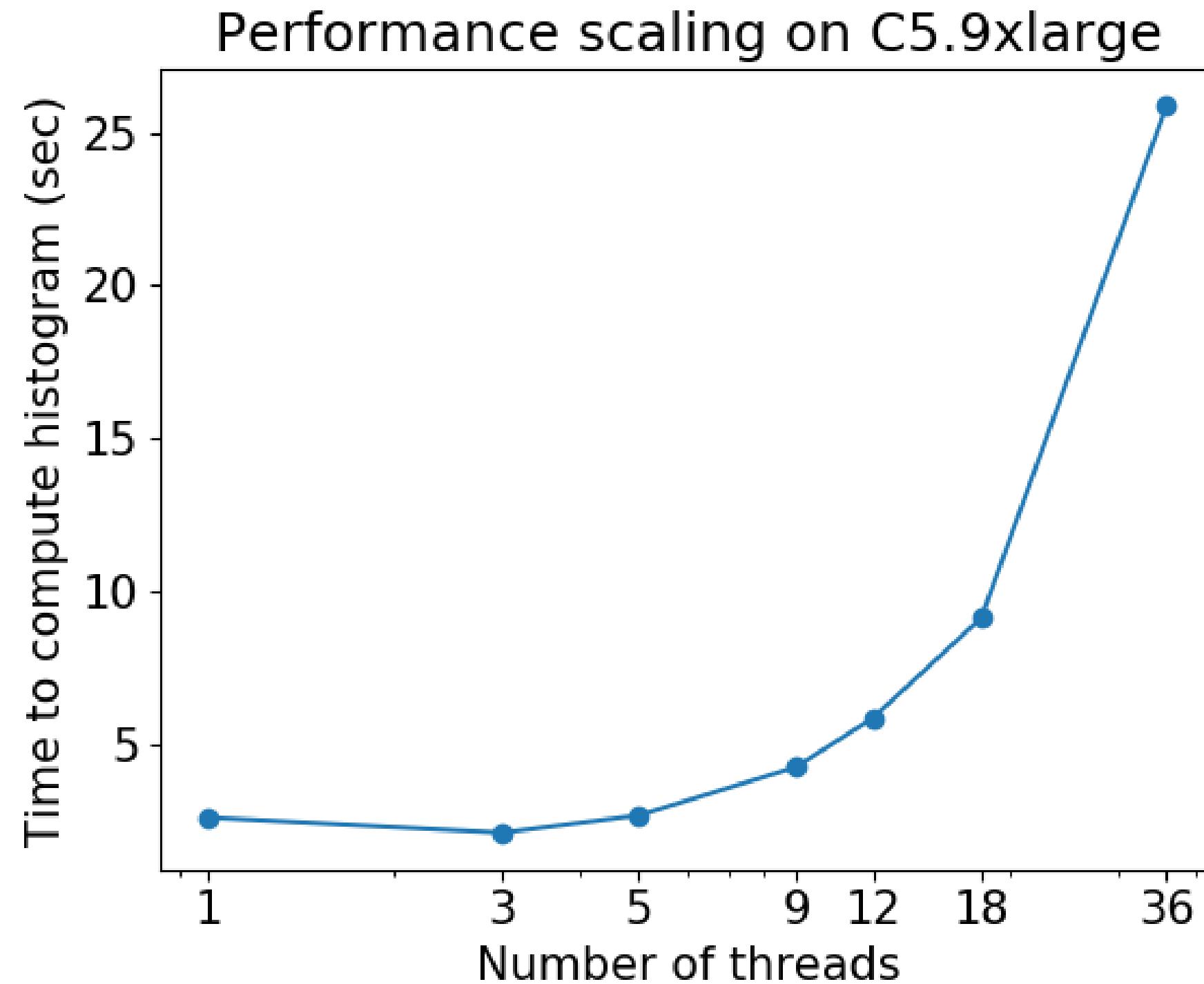
Growth of XGBoost project, in SLOC



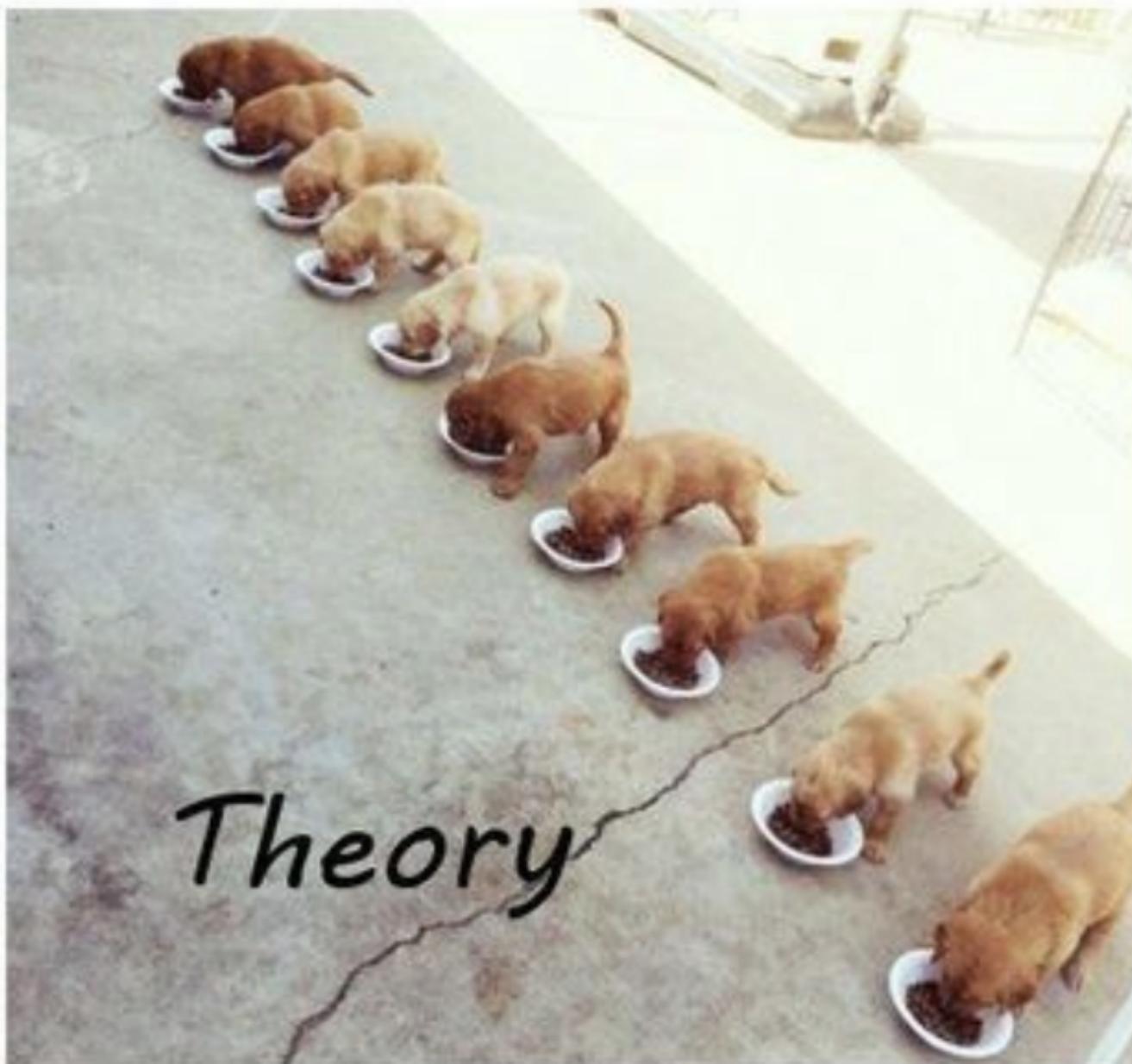
# GROWING PAINS



# OCT. 2018: PERFORMANCE BUG DISCOVERED



# Multithreaded programming



# JUNE 2019: SOMEONE SENT US A FIX, BUT...

	Abalone	Letters	Mortgage	Higgs1m	Airline	MSRank
BEFORE	10.0	144.6	53.4	123.5	133.2	332.2
AFTER	0.7	10.3	18.3	15.5	55.3	99.1
Difference, times	14.3	14.0	2.9	8.0	2.4	3.4

+1,341 -817 

# JUNE 2019: SOMEONE SENT US A FIX, BUT...



It's hard to  
understand  
their code

BUT they  
worked SO  
HARD!!

"No" would be too  
uncomfortable. I  
wanna be nice

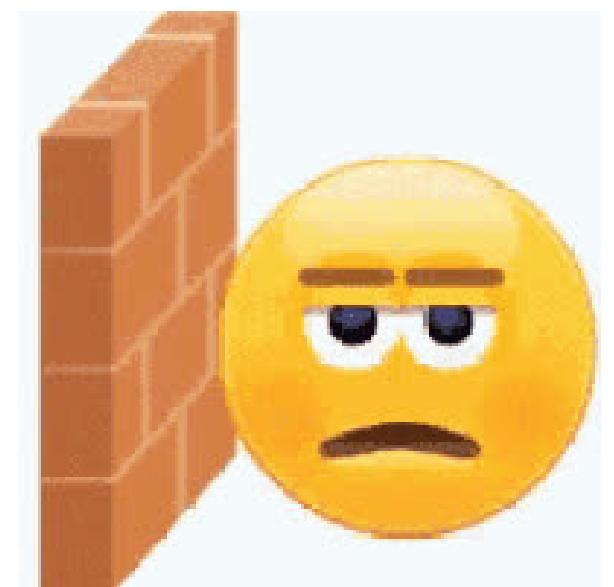


**SEVERAL  
MONTHS  
LATER...**

# NOV 2019: AT IMPASSE

“Those like CreateTasksForXXX have to go. I can't allocate time to do a full clean up for CPU Hist. I might prefer maintainability over performance, after banging my head on the monitor, I started to think about reverting the CPU optimization. WDYT?”

<https://github.com/dmlc/xgboost/pull/5005>



# NOV 2019: AT IMPASSE

“Let's revert for now and we'll try to incorporate some performance improvements in a later date. The state of the current 'hist' code is such that most developers hardly understand it at all and thus slowing down further progress.”

<https://github.com/dmlc/xgboost/pull/5005>



# NOV 2019: AT IMPASSE

“After [the pull request], the 'hist' updater became quite unwieldy and currently few developers understand it. In particular, CreateTasksForXXX functions have created multiple layers of indirection, making it unclear which objects are being computed/modified.”

# AFTERMATH

---

Slight awkward moment when the contributor found out we reverted their contribution



Fortunately they understood our decision and even submitted a replacement pull request that was much more legible.



As a result, we were able to start making further improvements

# LESSON LEARNED

---

Code legibility and clarity are just as important as performance.

Open source projects should consider long-term impact of the new proposed changes

Sometimes you have to say "no"

# STORY SO FAR

---

Software is awesome

We ask software lots of things

Complexity will come and bite you

BUT... there is a way out



# GOOD HABITS GO A LONG WAY

---

I will teach you good habits you can start practicing today

Some of the habits will **help your senior project!**

Avoid pitfalls and mistakes that'd derail your project  
Complete your project with high-quality code and a peace of mind

# #-1. START EARLY

Do not procrastinate!

Earlier you start, you can rescue yourself more easily out of mistakes and troubles.

Rushing for deadlines will make you sloppy

# #0. EXPECT TROUBLE

Fact: all software have **bugs**. So does your code.

Prepare for possible delays.

“Schedule padding” / “Underpromise, Overdeliver”

# #1. START SMALL, ADD SMALL

Adding lots of code in a short time period is a recipe for disaster

Add a little bit of code (a function or a class) at a time.

Then:

- Test the function/class by itself (later discussed)
- Let it rest a bit and review (\*)
- Save it to a version control system (next slide)
- Repeat

(\*) It's best to do code review in a pair or a group. Multiple eyes can uncover lots of issues

## #2. VERSION CONTROL

Use a version control system, e.g. [Git](#). Also back up your code to [GitHub](#)

Git helps you in multiple ways:

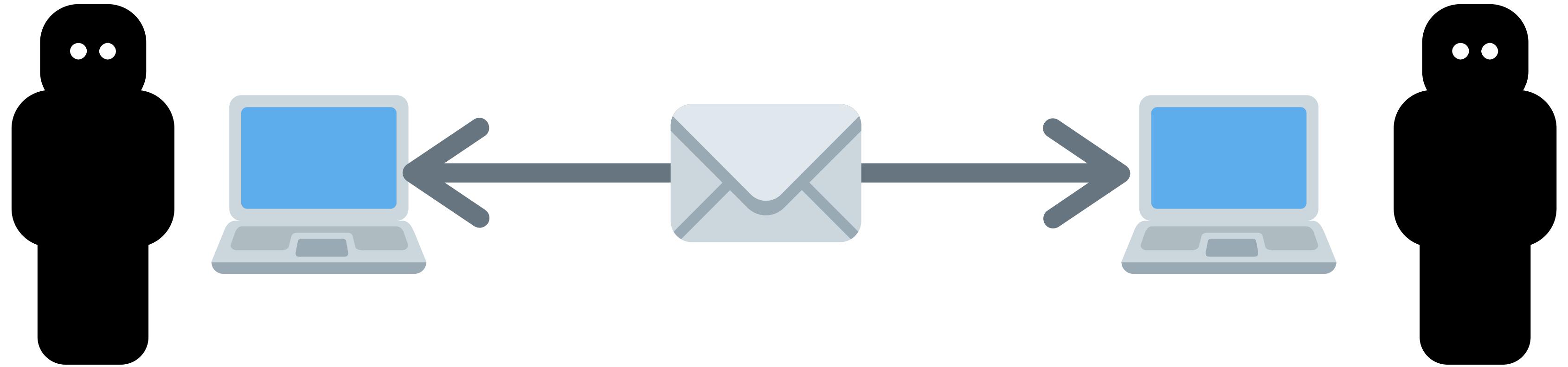
- Collaboration
- Documentation of changes over time / work log
- Ability to roll back changes

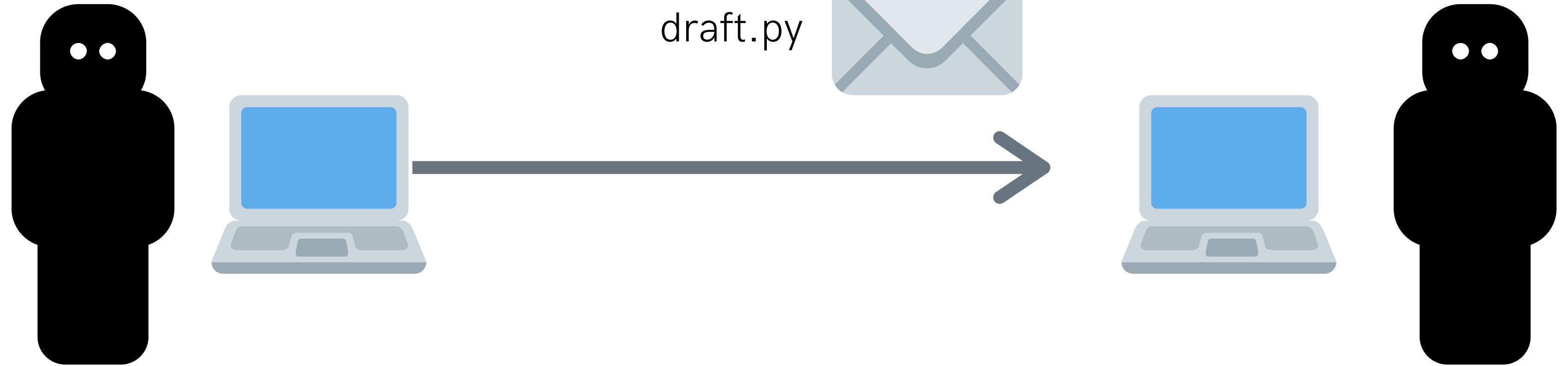
# Collaboration

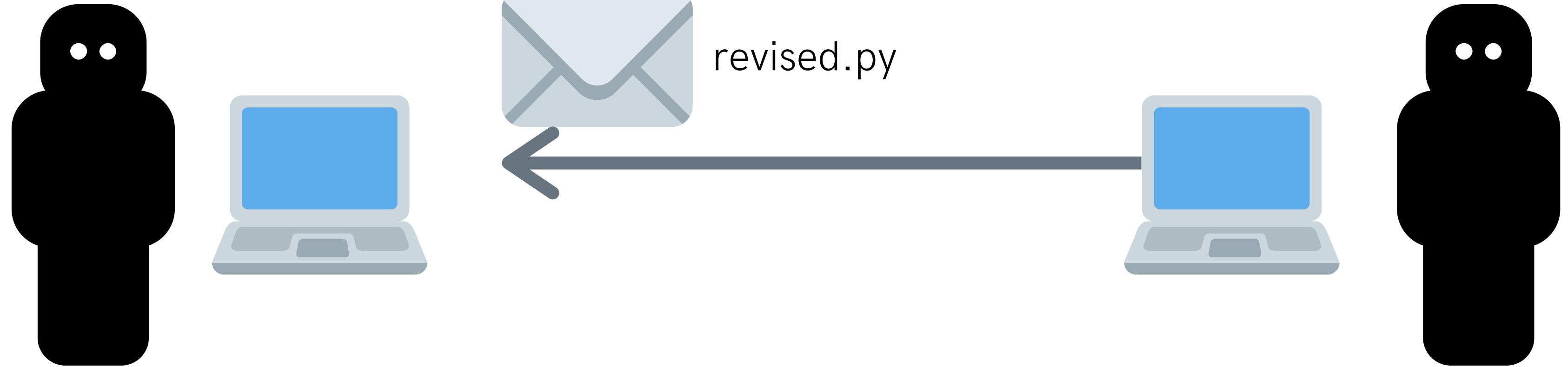


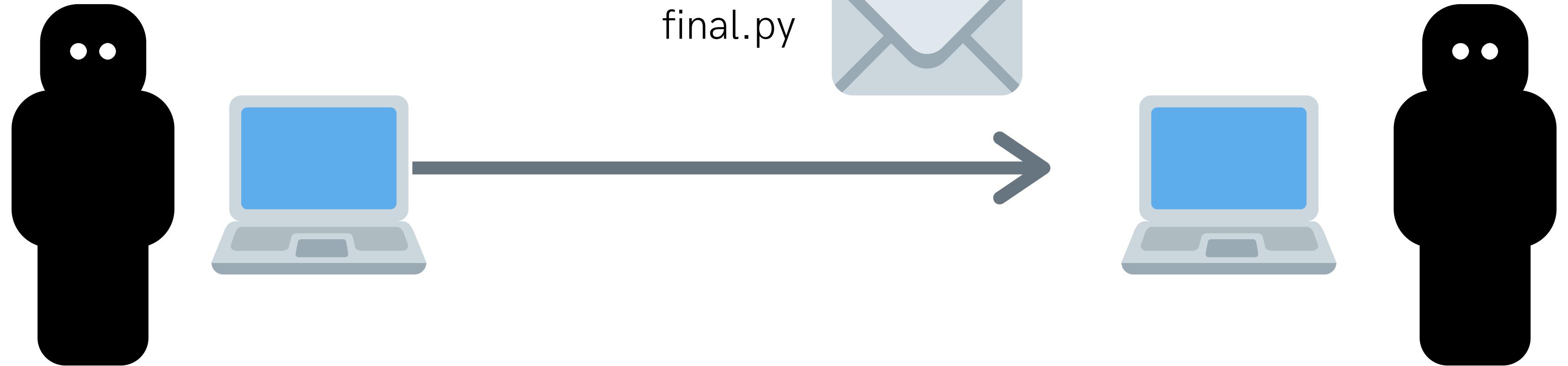
Meeting in a lab may not always be desirable

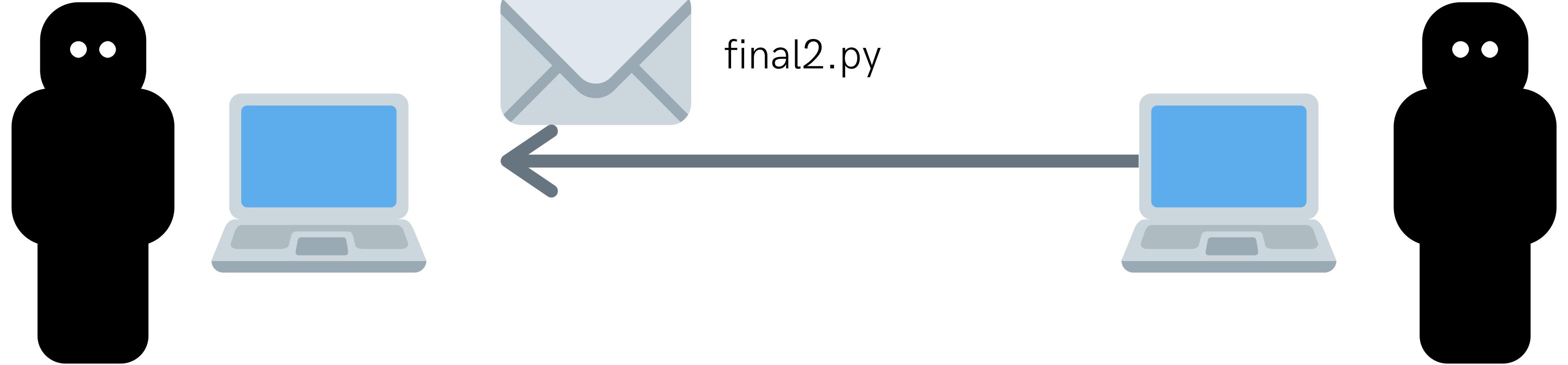
- Lab might be full
- Hard to find time that work for 2+ people
- Social distancing requirement

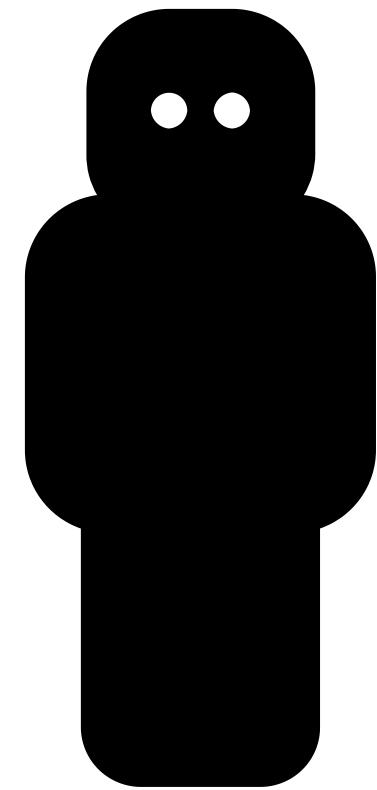




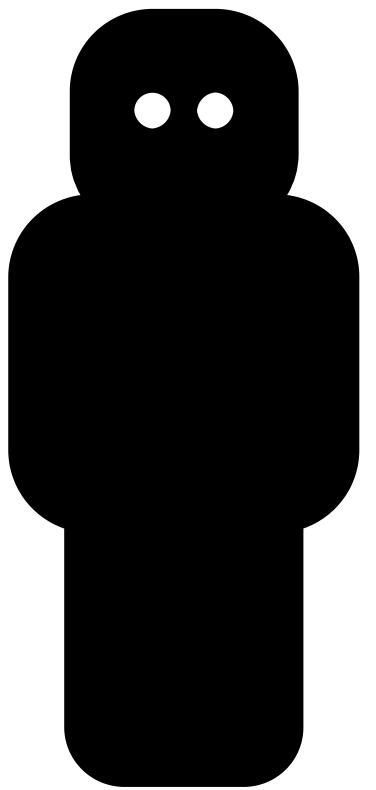


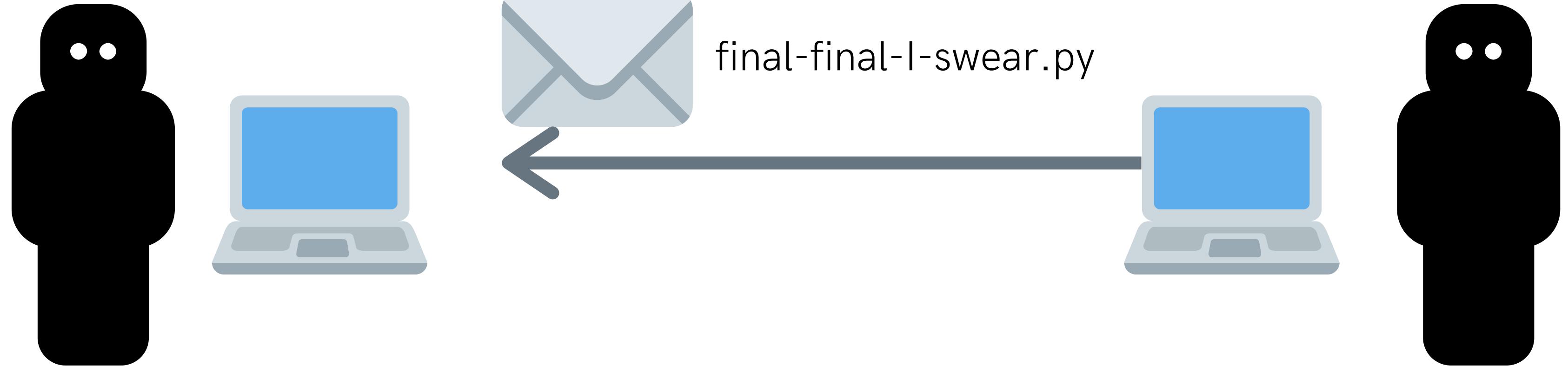


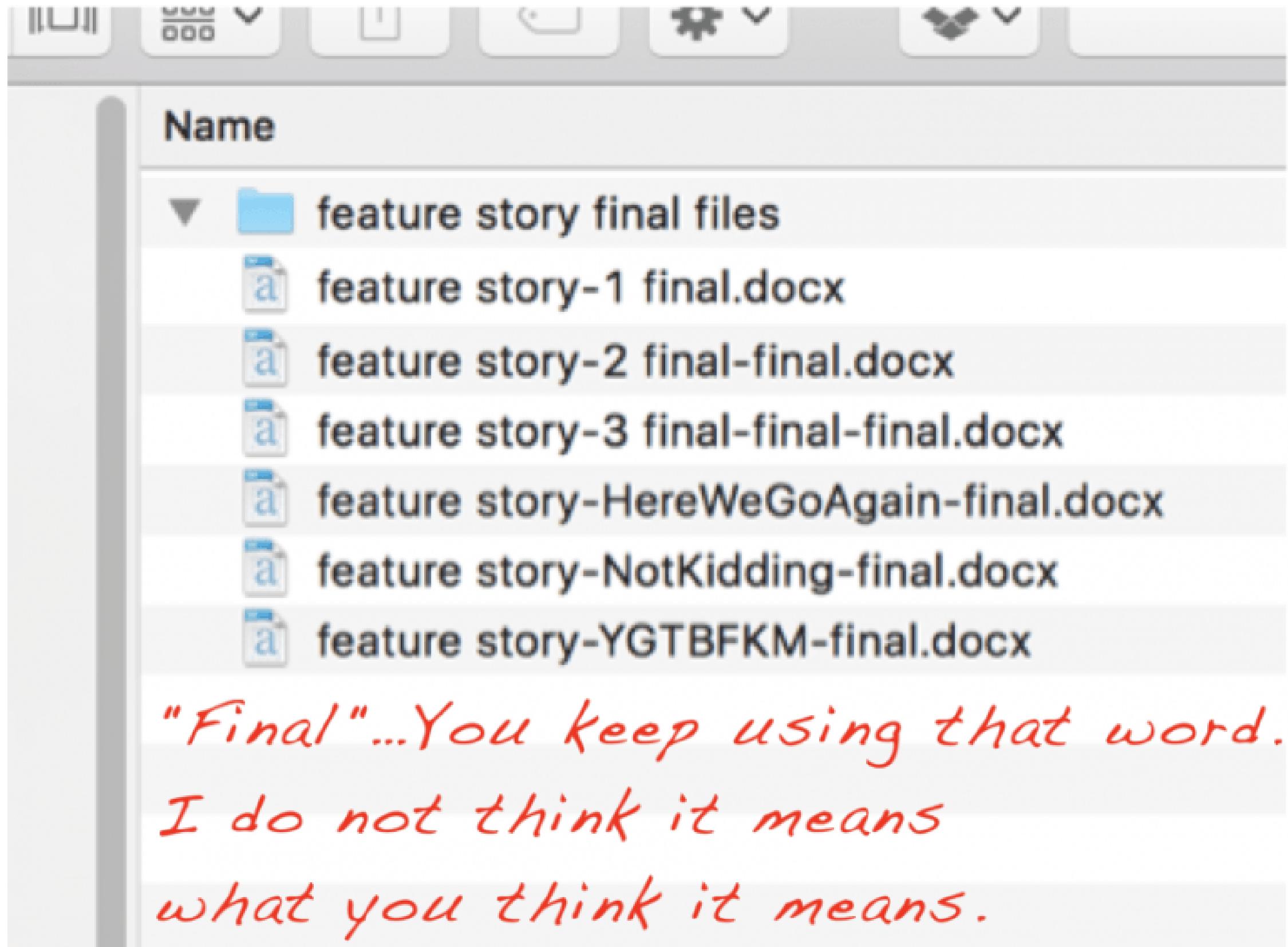




really-final.py

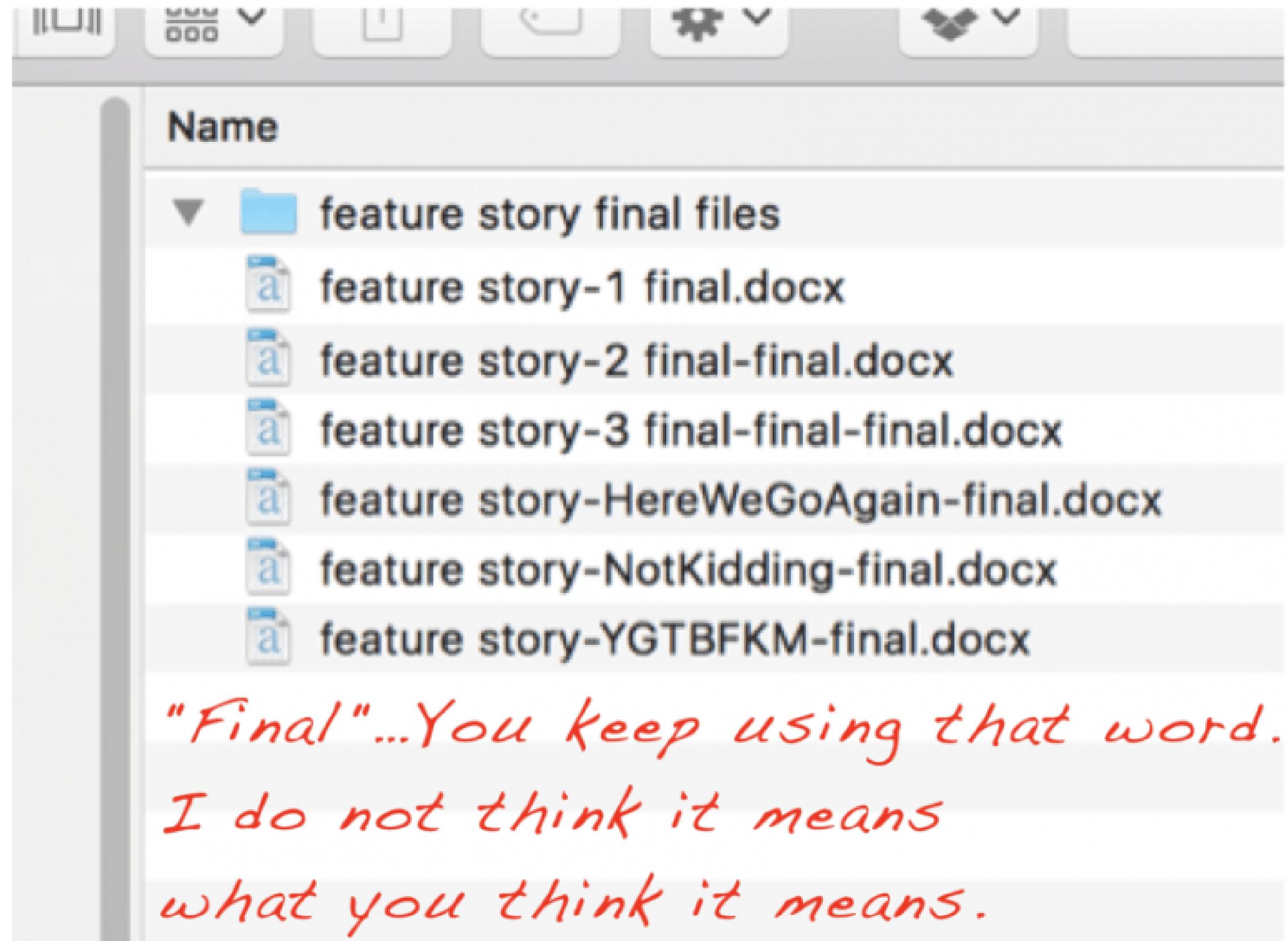






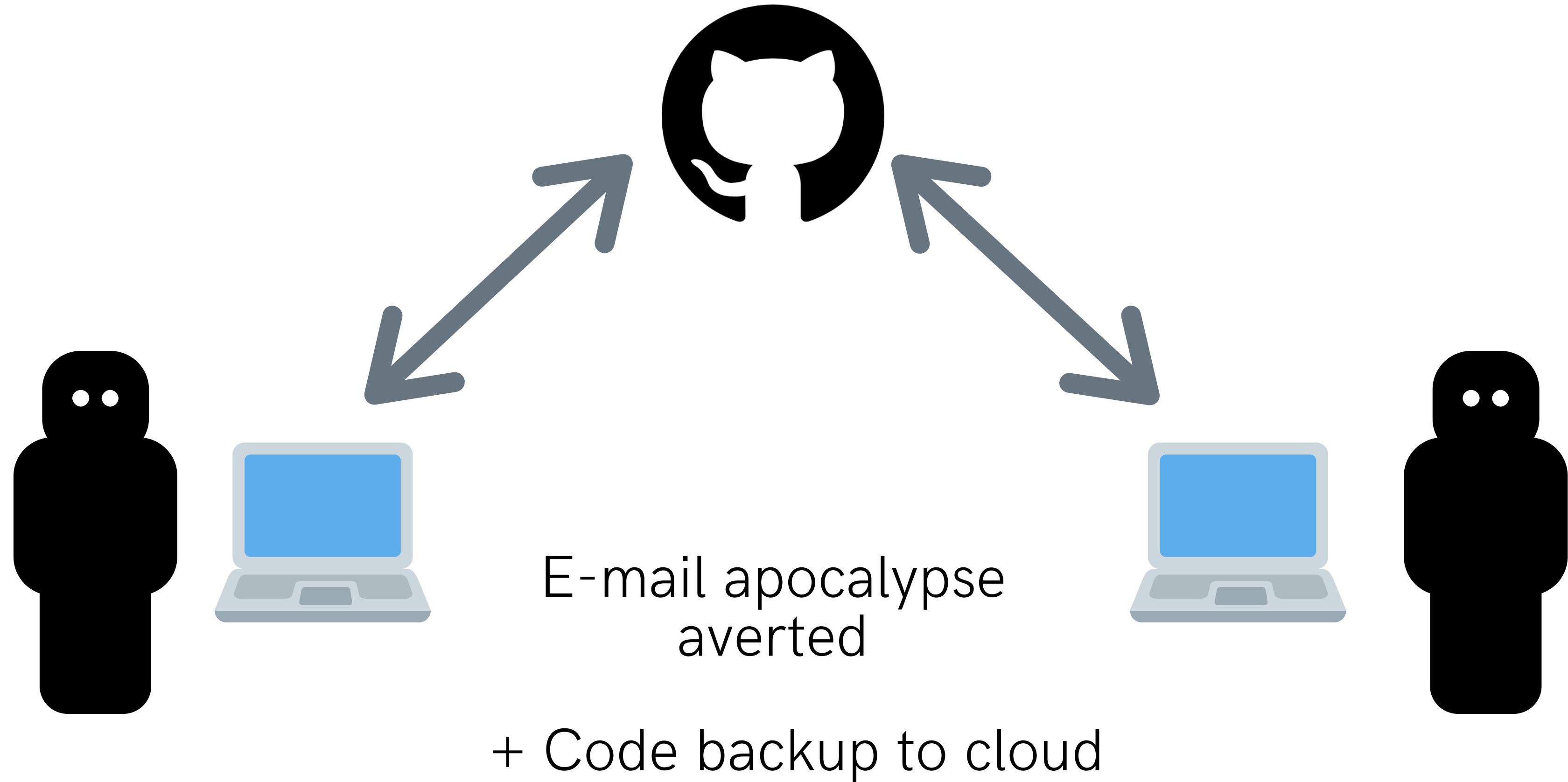
<https://doctorfreelance.com/file-naming-conventions/>

Too many e-mails  
Too many files  
Not trackable  
Might submit wrong version (!!)



<https://doctorfreelance.com/file-naming-conventions/>

# Collaboration: Git(Hub) to the Rescue



# Collaboration: Git(Hub) to the Rescue

Track changes,  
without duplicating files

master ▾

- o Commits on Mar 19, 2020
  - Use consistent sigma value (sigma=1) for all subplots**  
 hcho3 committed on Mar 19
- o Commits on Dec 19, 2019
  - changes to data**  
 avinashbarnwal committed on Dec 19, 2019
  - changes to data**  
 avinashbarnwal committed on Dec 19, 2019

# Collaboration: Git(Hub) to the Rescue

## Code reviews and pull requests

### Use consistent sigma value (sigma=1) for all subplots #7

Merged avinashbarnwal merged 1 commit into `avinashbarnwal:master` from `unknown repository` on Mar 19

Conversation 0 Commits 1 Checks 0 Files changed 2 +8 -8

Changes from all commits ▾ File filter... ▾ Jump to... ▾ Review changes ▾

0 / 2 files viewed

Viewed ...

16 AFT/R/combined\_assignment.R

```
@@ -249,10 +249,10 @@ distribution.list = list(gaussian=list(uncensored_l=loss(type="uncensored",t.low
249 left_h=hessian(type="lef
t",t.lower=-Inf,t.higher=20,sigma=1,y.hat=2**
250 (seq(1,x.lim,length=n.points)),dist='normal'),
right_h=hessian(type="ri
ght",t.lower=60,t.higher=Inf,sigma=1,y.hat=2**
251 (seq(1,x.lim,length=n.points)),dist='normal'),
interval_h=hessian(type=
"interval",t.lower=16,t.higher=200,sigma=1,y.hat=2**
252 (seq(1,x.lim,length=n.points)),dist='normal')),
252 +
logistic=list(uncensored_l=loss(type="uncensored",t.lower=100,t.h
igher=100,sigma=sqrt(pi^2/6),y.hat=2**
253 (seq(1,x.lim,length=n.points)),dist='logistic'),
left_l=loss(type="left",t.lower=--
```

# Collaboration: Git(Hub) to the Rescue

Roll back changes  
as necessary

My program crashes here.... →

... but not here →

Git history



# Collaboration: Git(Hub) to the Rescue

Roll back changes  
as necessary

Roll back to this version

Git history



# Collaboration: Git(Hub) to the Rescue

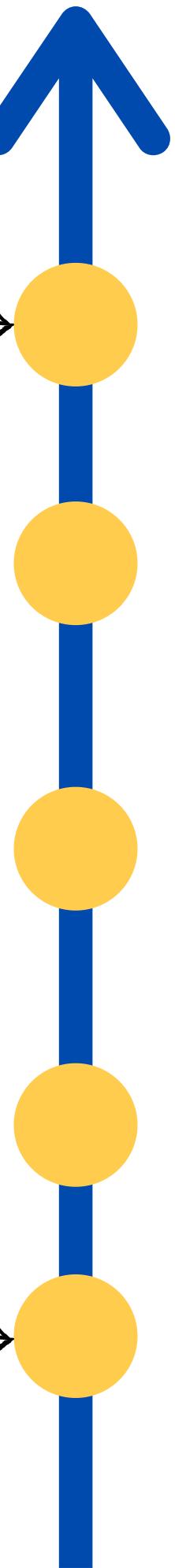
Git history

Find out which  
change (commit)  
introduce a bug  
(git bisect)

My program crashes here.... →

Somewhere a bug was introduced

... but not here →



# Collaboration: Git(Hub) to the Rescue

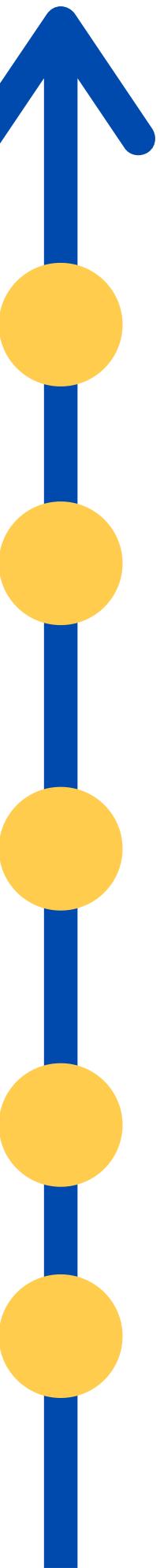
Git history

Find out which  
change (commit)  
introduce a bug  
(git bisect)

Exact commit that introduced the bug →

My program crashes here.... →

... but not here →



## Collaboration: Git(Hub) to the Rescue

Sign up at [GitHub.com](https://GitHub.com) and learn Git today! It's free!

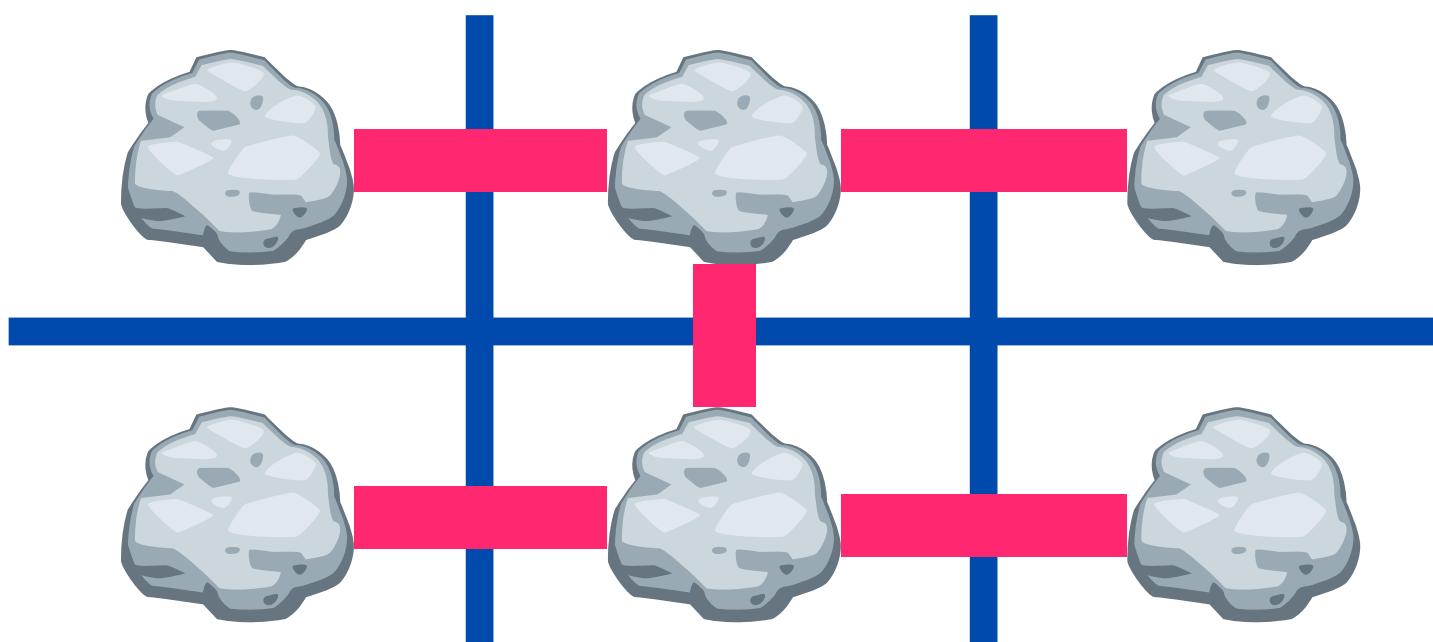
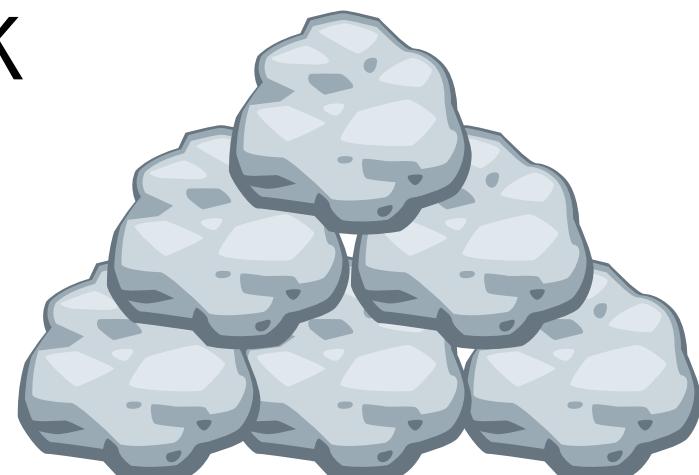
This will pay off immensely

Note: Make sure to use [private repository](#) to store your senior project

# #3. MODULAR DESIGN

Organize your code into multiple **modules**

- Quite difficult to get it right first time
- Advice #1 definitely helps
- Get better with experience, e.g.
  - Open source
  - Internship
  - Work



# #4. UNIT TESTING

Add small bit of code at a time, and then test it

- **Unit test:** Given an input, does a function produce a correct output?
  - If not, your code has a bug; fix it!

# #4. UNIT TESTING

```
def area_of_rectangle(width, height):  
    return width * height  
  
def test_rectangle():  
    assert area_of_rectangle(1, 4) == 4  
    assert area_of_rectangle(4, 1) == 4  
    assert area_of_rectangle(2, 2) == 4  
    assert area_of_rectangle(2.5, 1.6) == 4
```

# #4. UNIT TESTING

Add small bit of code at a time, and then test it

- **Mocking:** Create fake parts that interact with a given module

Menu 1

Menu 2

Menu 3

*Placeholder for Menu 1*

# #4. UNIT TESTING

Add small bit of code at a time, and then test it

- **Mocking:** Create fake parts that interact with a given module



*Placeholder for Menu 2*

# #4. UNIT TESTING

Add small bit of code at a time, and then test it

- **Mocking:** Create fake parts that interact with a given module



*Placeholder for Menu 3*

# #4. UNIT TESTING

Add small bit of code at a time, and then test it

- Tools
  - Pytest (Python)
  - Jest (JavaScript)
  - Google Tests (C++)
  - JUnit (Java)

# CONCLUSION

Good habits let you avoid worst pitfalls of software complexity.

# CONCLUSION

Good habits let you avoid worst pitfalls of software complexity.

Experience with real-world software trains your sense, to smell "bad code" from miles away

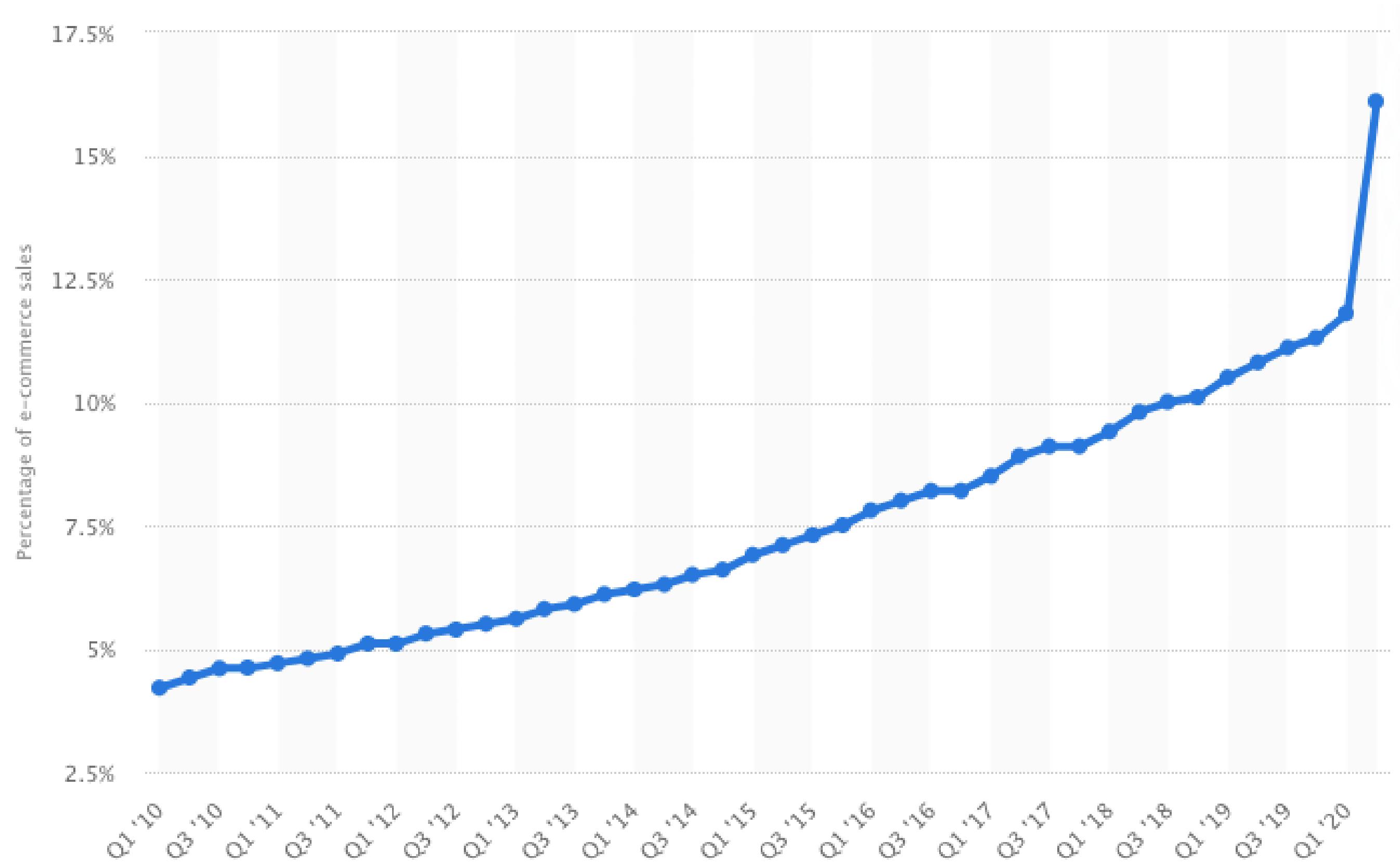
- Consider getting an **internship** / contributing to an **open-source project**
- You should avoid mistakes, but mistakes are often fastest way for you to learn

FYI: We are hiring interns

# **YEAR 2021: DIFFICULT TIME, ALSO WITH OPPORTUNITIES**

Economy still under effect of the pandemic

Keep in mind of the opportunities



Quarterly share of e-commerce sales of total U.S. retail sales

<https://www.statista.com/statistics/187439/share-of-e-commerce-sales-in-total-us-retail-sales-in-2010/>



Small businesses (\*) comprise 99.5% of firms and employ 52% of Americans (\*\*). They took the brunt of the current recession. Many are building up IT infrastructure and establishing online presence.

(\*) < 500 full-time employees

(\*\*) As of 2019. "Business Employment Dynamics," U.S. Bureau of Labor Statistics

<https://www.bls.gov/bdm/bdmfirmsize.htm>

Small businesses (\*) comprise 99.5% of firms and employ 52% of Americans (\*\*). They took the brunt of the current recession. Many are building up IT infrastructure and establishing online presence.

=> Software is expanding its reach in the national economy. The pandemic will eventually pass, but the trend will continue.

(\*) < 500 full-time employees

(\*\*) As of 2019. "Business Employment Dynamics," U.S. Bureau of Labor Statistics

<https://www.bls.gov/bdm/bdmfirmsize.htm>

Small businesses (\*) comprise 99.5% of firms and employ 52% of Americans (\*\*). They took the brunt of the current recession. Many are building up IT infrastructure and establishing online presence.

=> Software is expanding its reach in the national economy. The pandemic will eventually pass, but the trend will continue.

=> We need more software developers.

(\*) < 500 full-time employees

(\*\*) As of 2019. "Business Employment Dynamics," U.S. Bureau of Labor Statistics

<https://www.bls.gov/bdm/bdmfirmsize.htm>

## Advice

- Be a good self-learner
- Be a good listener
- Be coachable
- Be willing to wear multiple hats (if needed)
- Do not box yourself. Step outside of the comfort zone.
- Above all, **pursue excellence in coding**. The more experience you get, the better.

Feel free to reach out to me at  
**[phcho@nvidia.com](mailto:phcho@nvidia.com)**