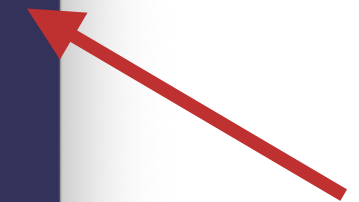
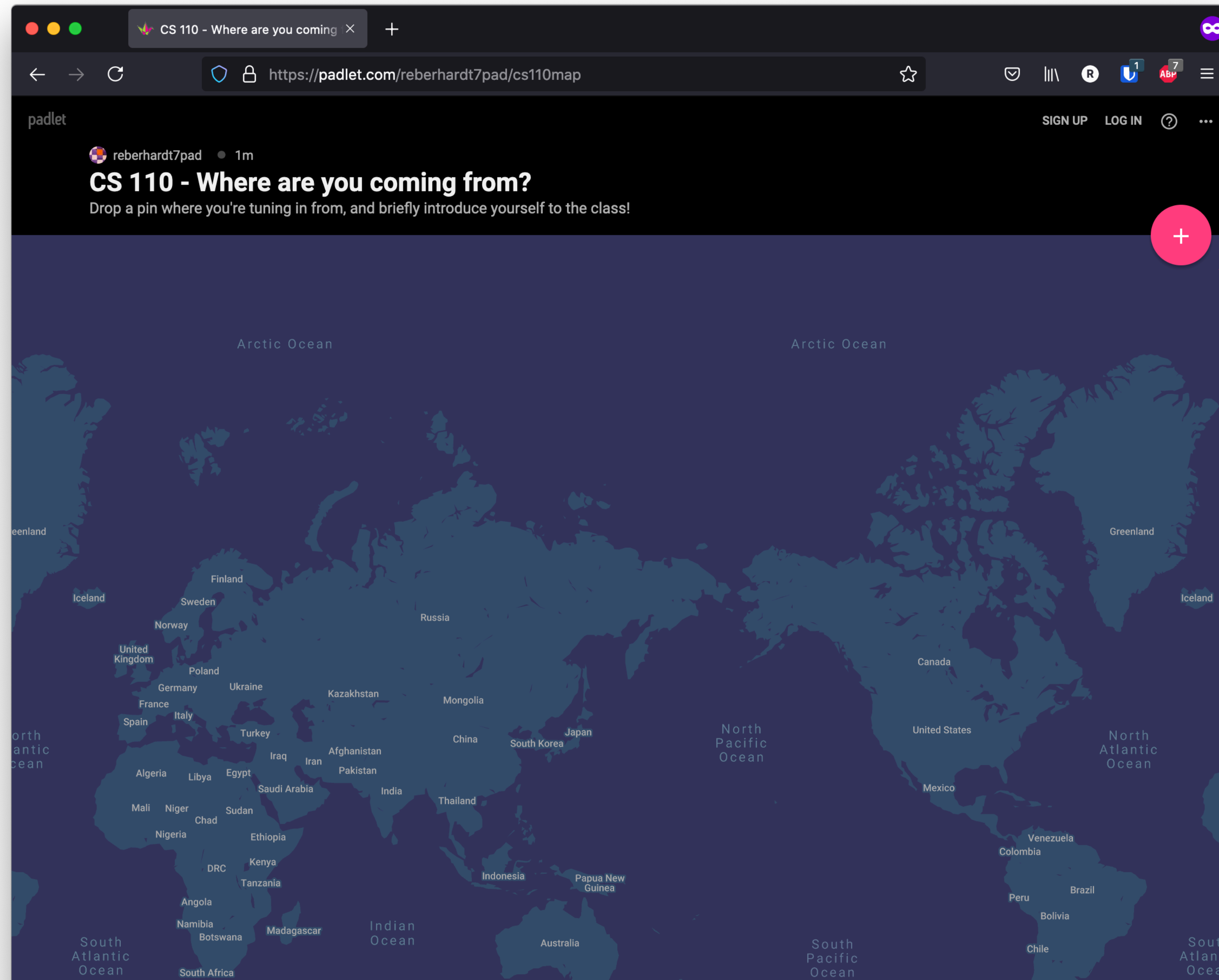


Introduce yourself to the class! 🙌

<https://padlet.com/reberhardt7pad/cs110map>



Click here 😊

Welcome to CS 110 🙌

Ryan Eberhardt
June 21, 2021

Today's lecture:

Today's lecture: a boring sandwich 🥪

CS 110: Principles of Computer Systems

Principles and practice of engineering of computer software and hardware systems. Topics include: techniques for controlling complexity; strong modularity using client-server design, virtual memory, and threads; networks; atomicity and coordination of parallel activities. Prerequisite: 107.

Terms: Aut, Win, Spr, Sum | **Units:** 3-5 | **UG Reqs:** GER:DB-EngrAppSci

Instructors: Cain, J. (PI) ; Cyrus, R. (PI) ; Eberhardt, R. (PI) ... [more instructors for CS 110](#) »

[Schedule for CS 110](#)

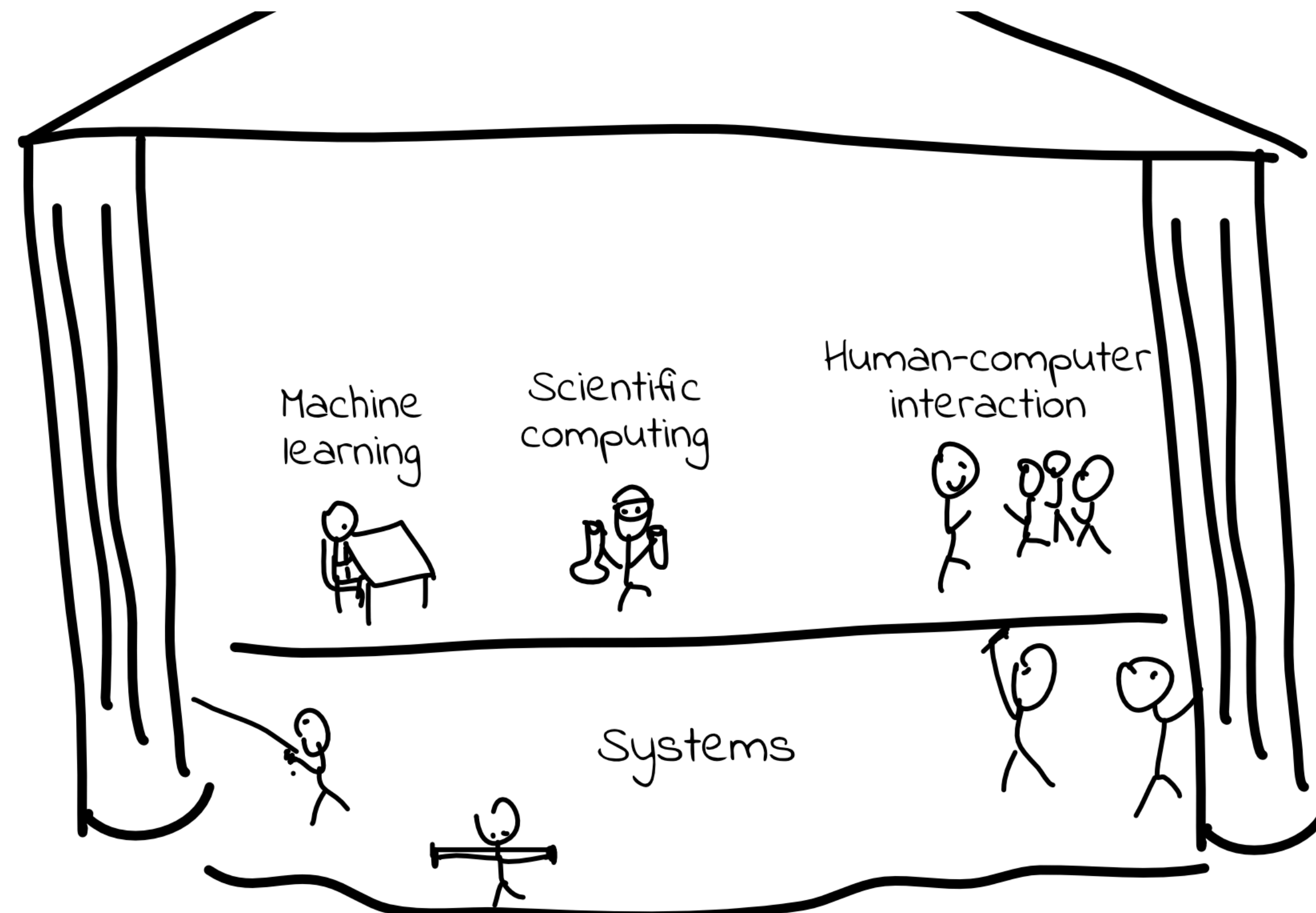
Today's lecture: a boring sandwich 🥪

- Interesting: What the heck is “systems” anyways?
- Boring: How is this class going to work?
- Interesting: What are filesystems, and how do they work?

What the heck is “systems” anyways?

Computer systems

- “Computer systems” is so broad and vague that it seems it could refer to anything involving computers
- I view systems as being about building the platform that all application software stands on



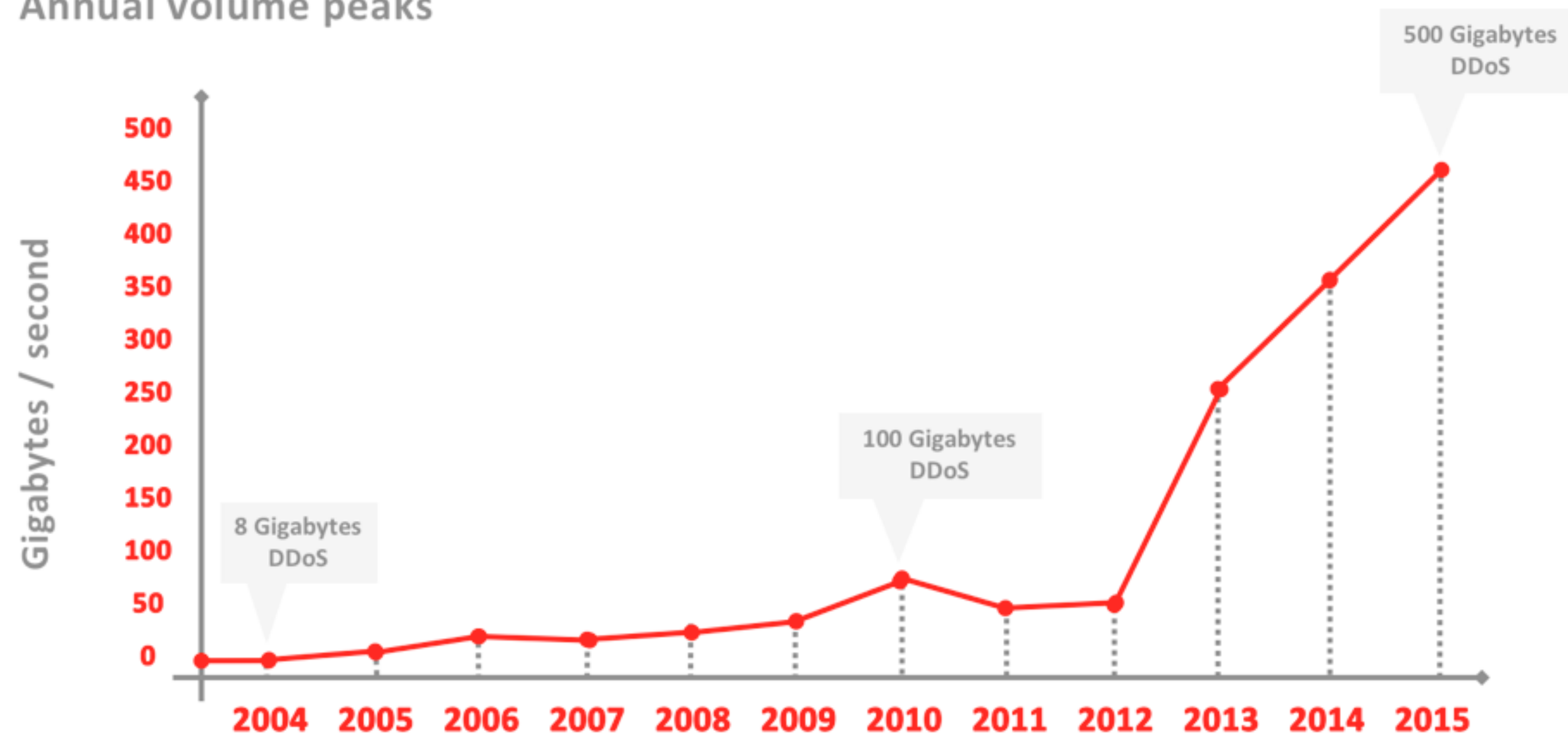
Computer systems

- “Computer systems” is so broad and vague that it seems it could refer to anything involving computers
- I view systems as being about building the platform that all application software stands on
- Systems people are the carpenters, masons, electricians, *and also* the architects and civil engineers
 - You’re designing a solution around high-level goals and tradeoffs
 - You’re also getting your hands dirty building it, fixing problems, etc

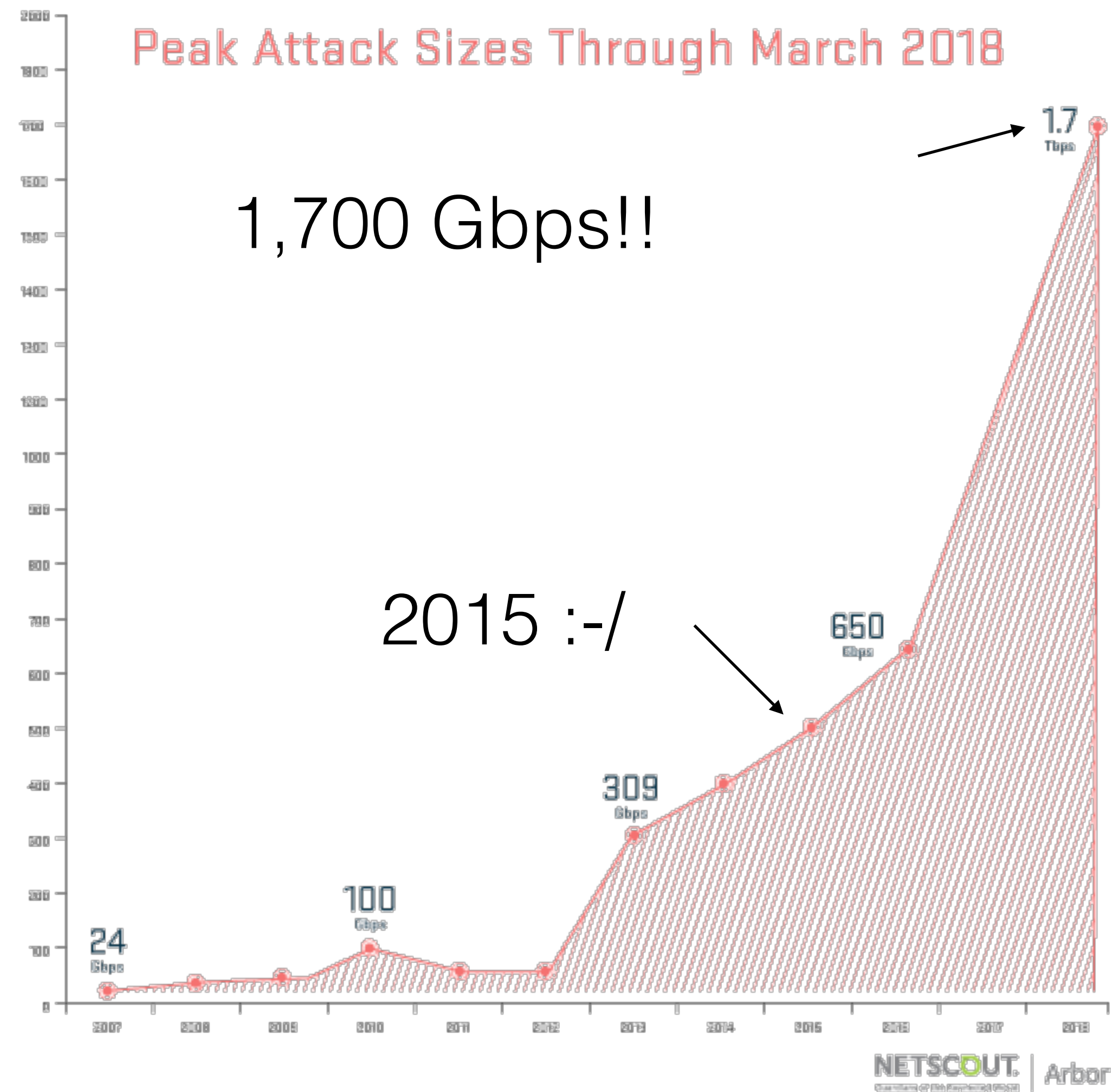
DDoS Attacks

DDoS attack evolution

Annual volume peaks

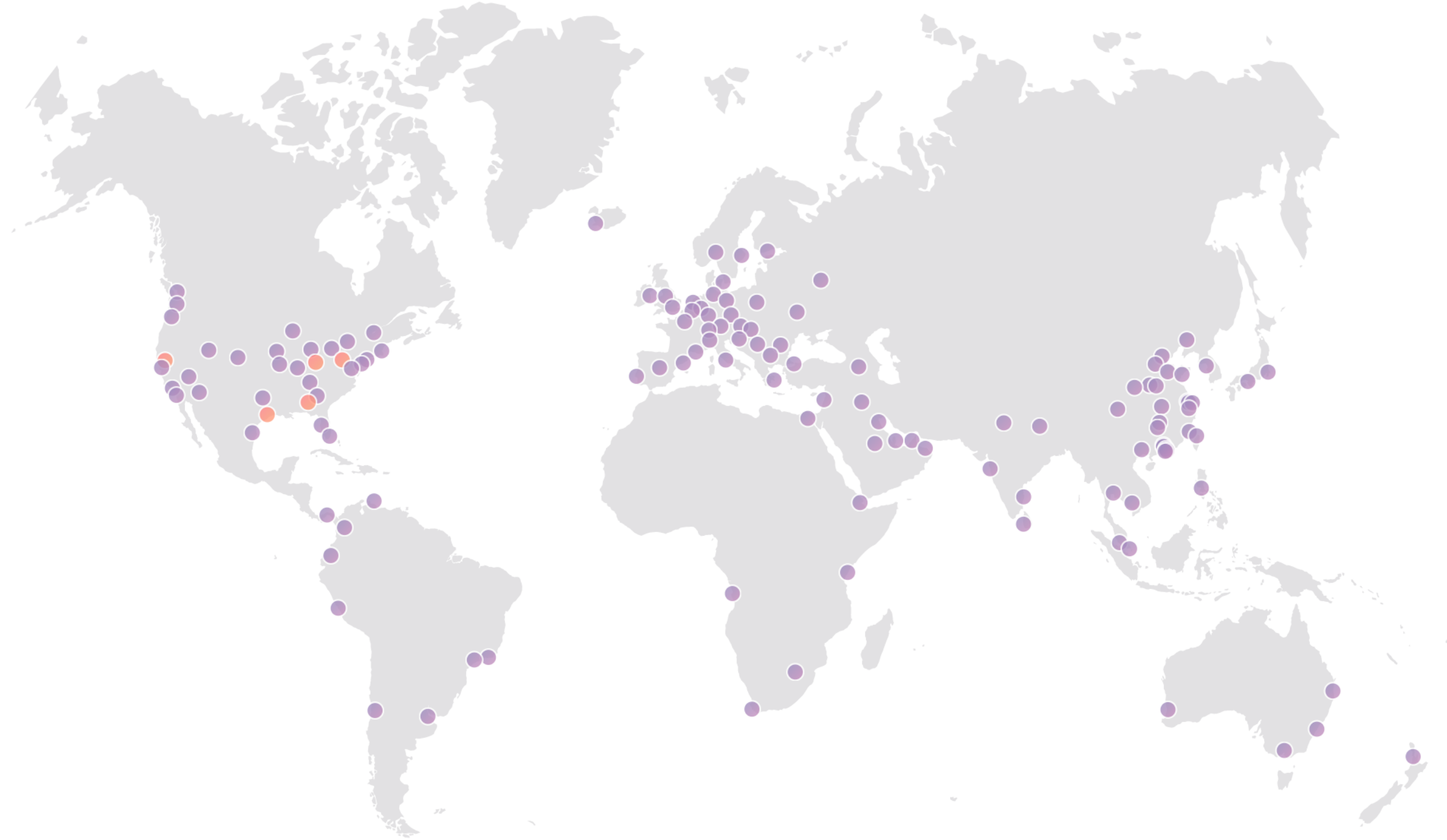


DDoS Attacks



Cloudflare

133 *datacenters*



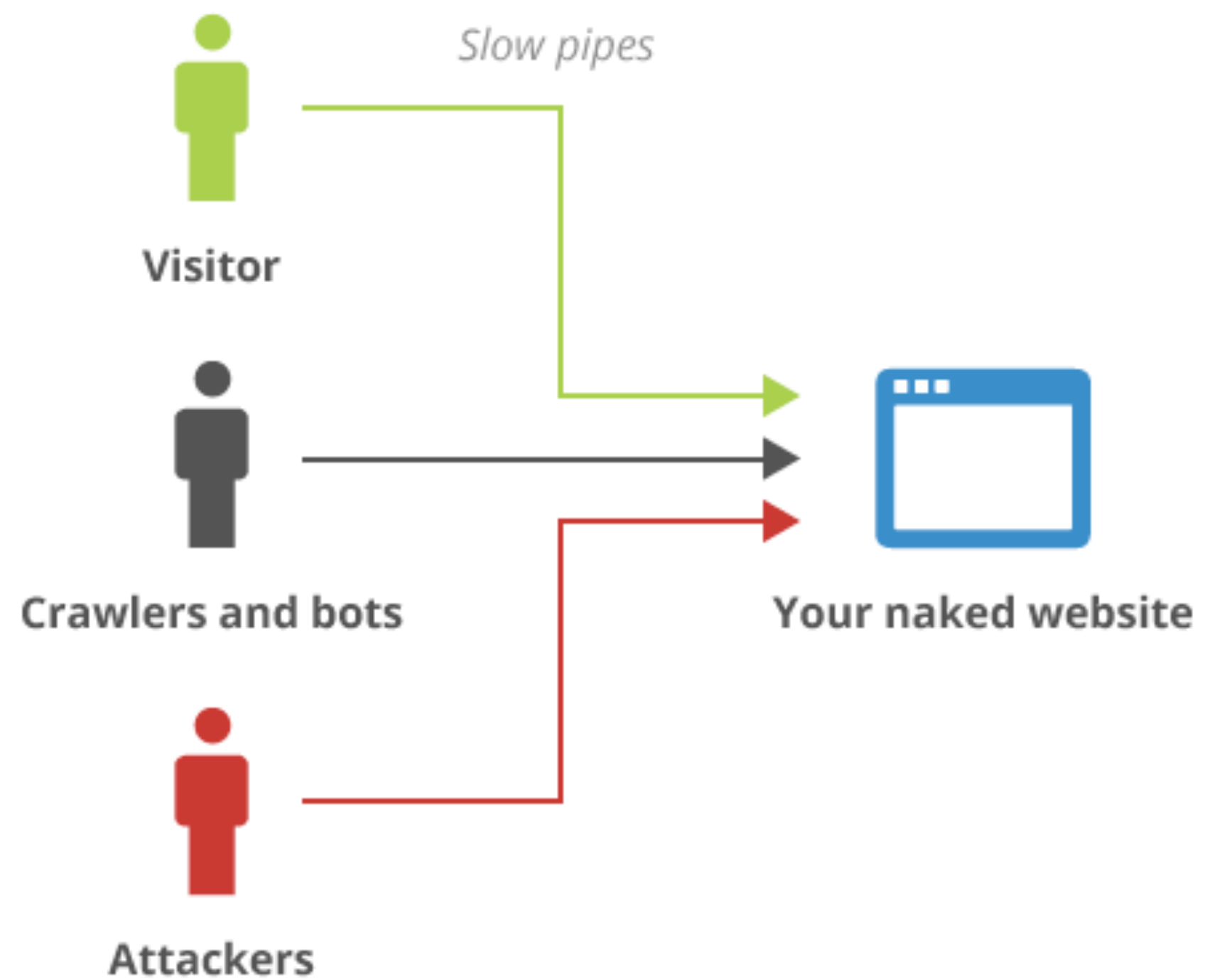
Cloudflare

133 *datacenters*

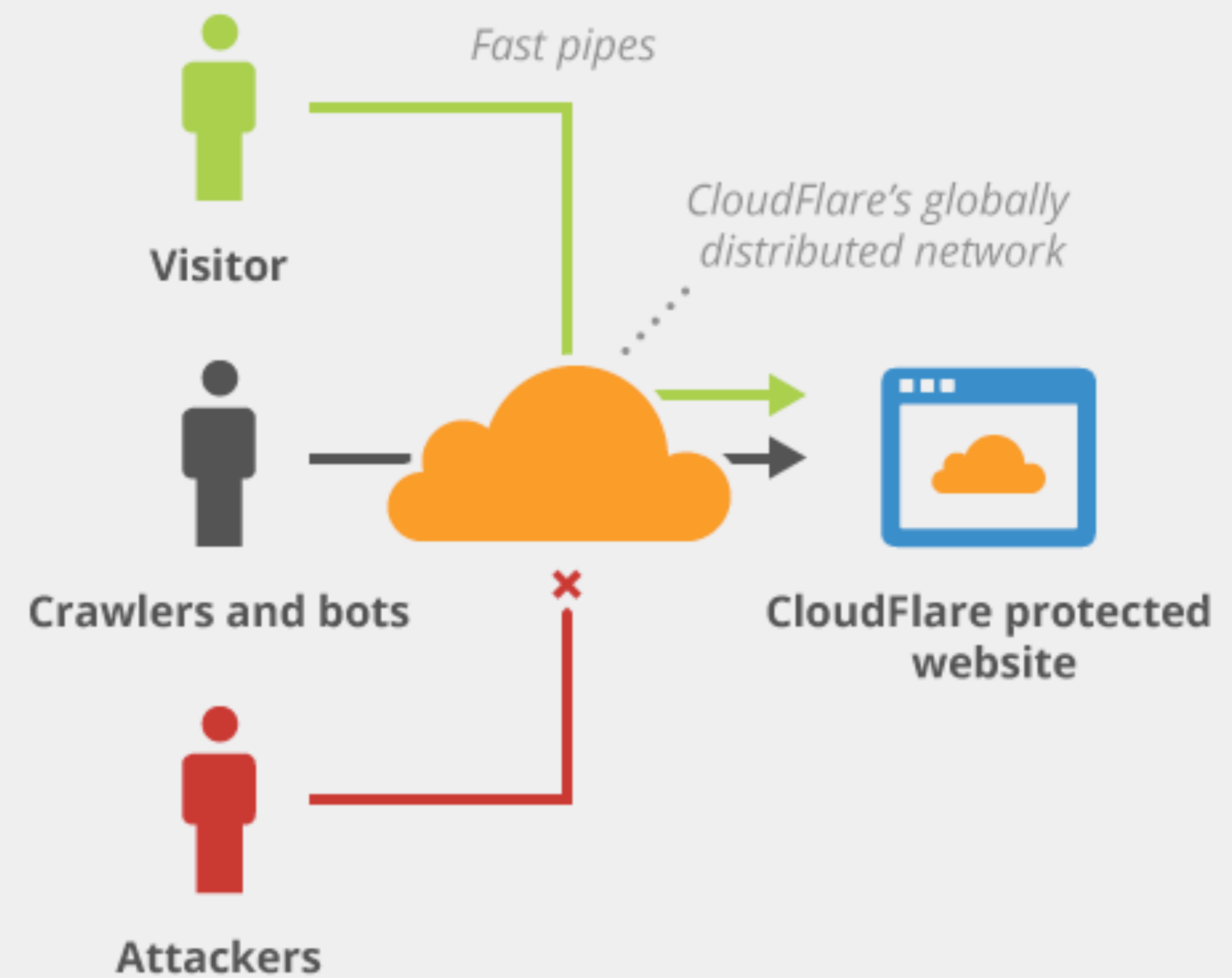


Cloudflare

Without CloudFlare



With CloudFlare



Cloudflare

<https://blog.cloudflare.com/>

Disney: Large-scale rendering

Without global illumination:



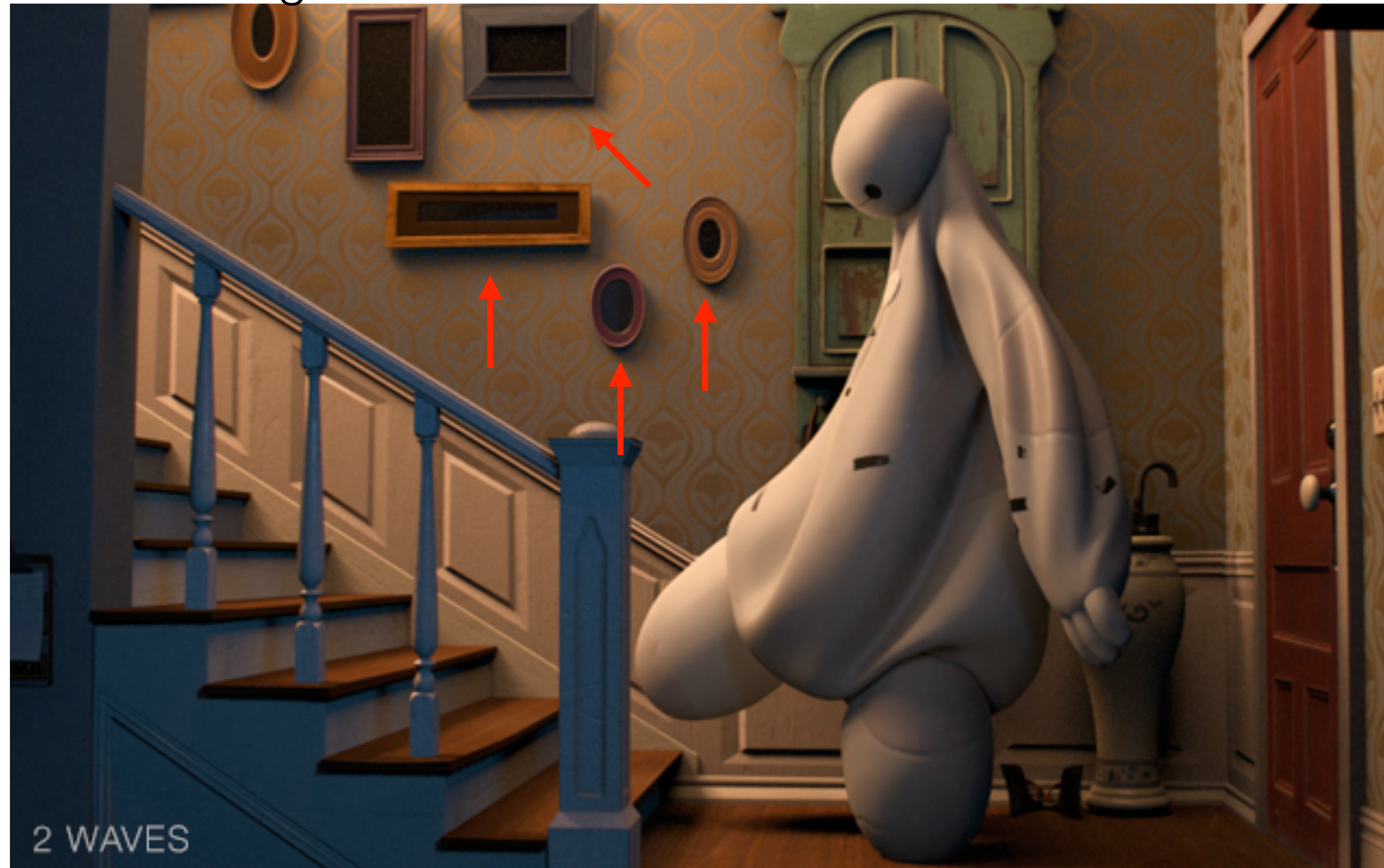
Disney: Large-scale rendering

With global illumination:



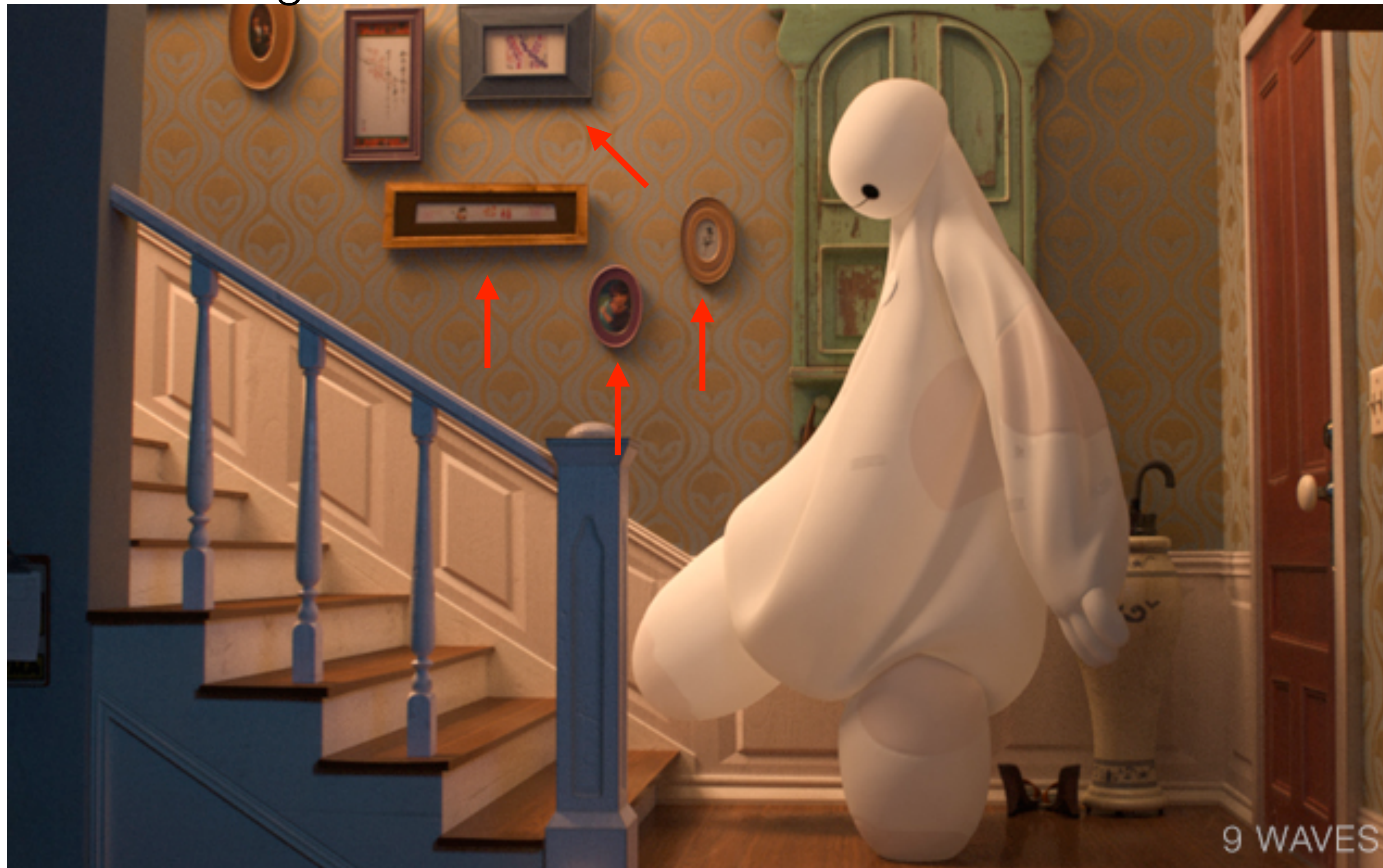
Disney: Large-scale rendering

Without global illumination:



Disney: Large-scale rendering

With global illumination:



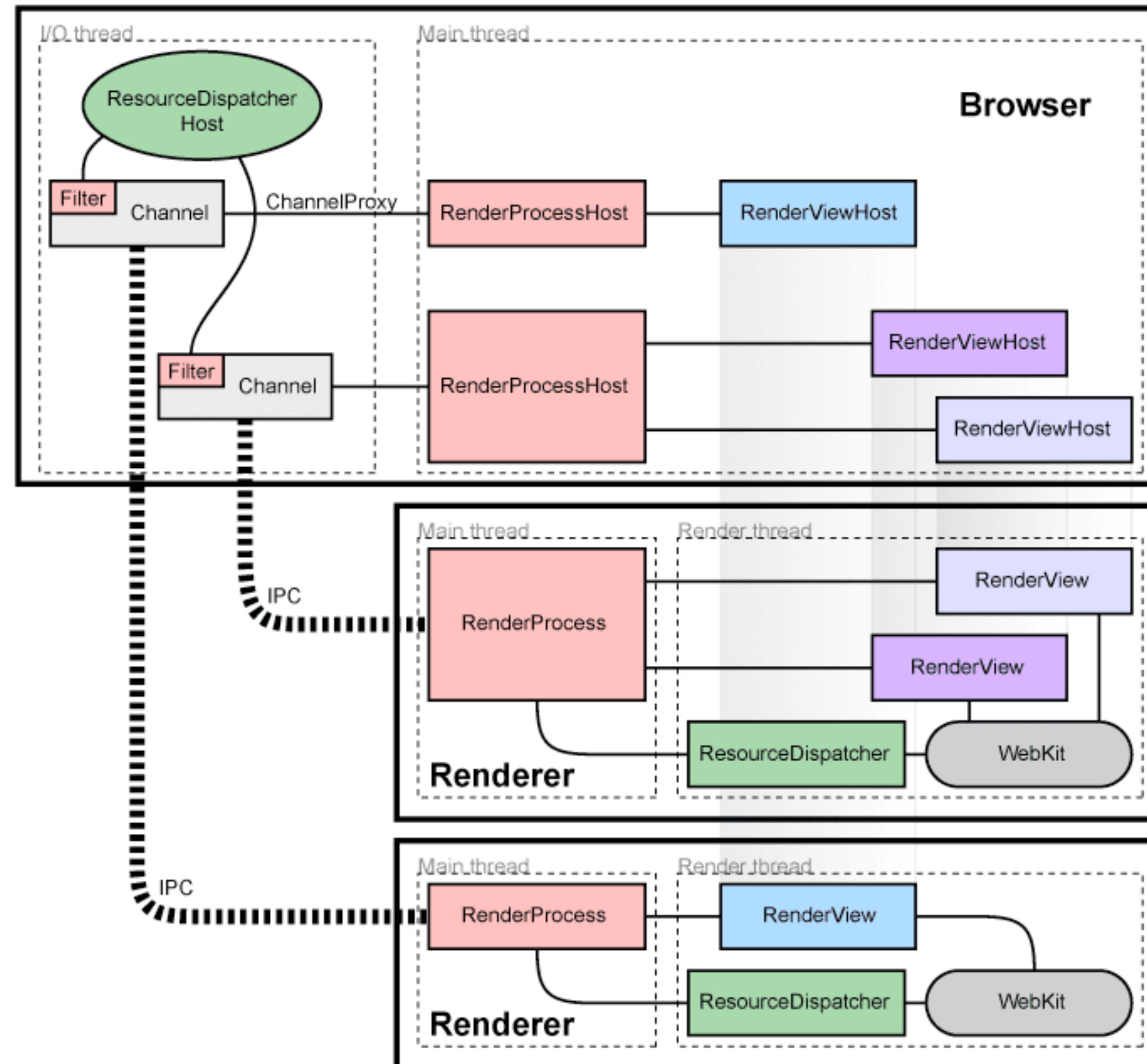
Disney: Large-scale rendering

- “San Fransokyo” contains 83,000 buildings, 260,000 trees, 215,000 streetlights, and 100,000 vehicles. City detail is based on assessor data from San Francisco
- Rendered in four geographically-distributed datacenters
- 55,000 CPU cores, 400 TB of memory
- Many system failures!

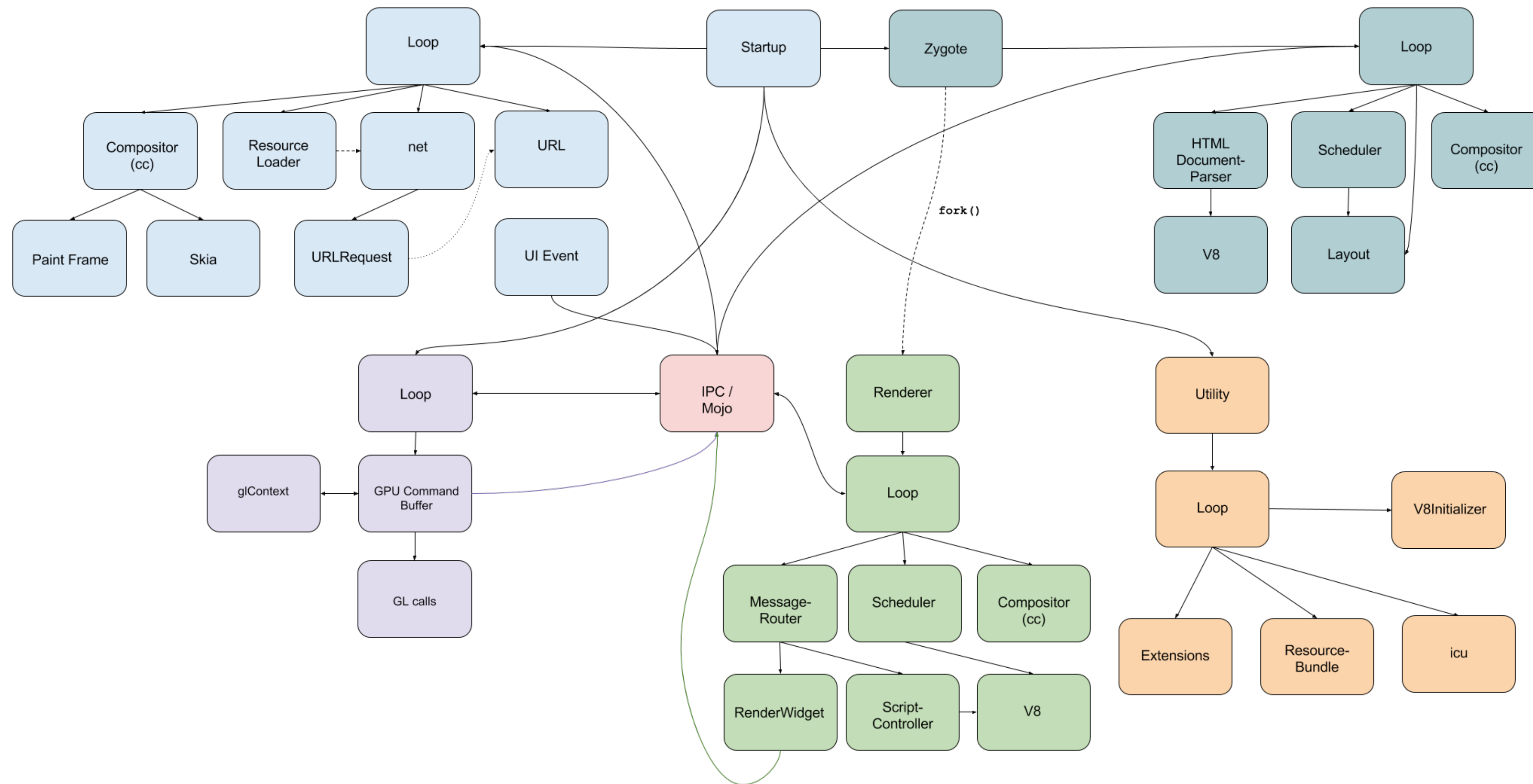
Google Chrome: More complex than you think!

- The modern browser is basically an operating system
- You can do almost anything in a browser:
 - Music synthesis, hardware MIDI interfaces: <https://musiclab.chromeexperiments.com/>
 - Render high quality graphics + physics simulations in real time: <http://madebyevan.com/webgl-water/>
 - Run Windows 95: <https://win95.ajf.me/>
- Security and isolation between sites is essential
 - A shady website you stumble upon should never have access to your email, bank, etc.

Google Chrome: More complex than you think!



Google Chrome: More complex than you think!



Today's lecture: a boring sandwich 🥪

- Interesting: What the heck is “systems” anyways?
- Boring: How is this class going to work?
- Interesting: What are filesystems, and how do they work?

CS 110: Principles of Computer Systems



Hello world! 🙌

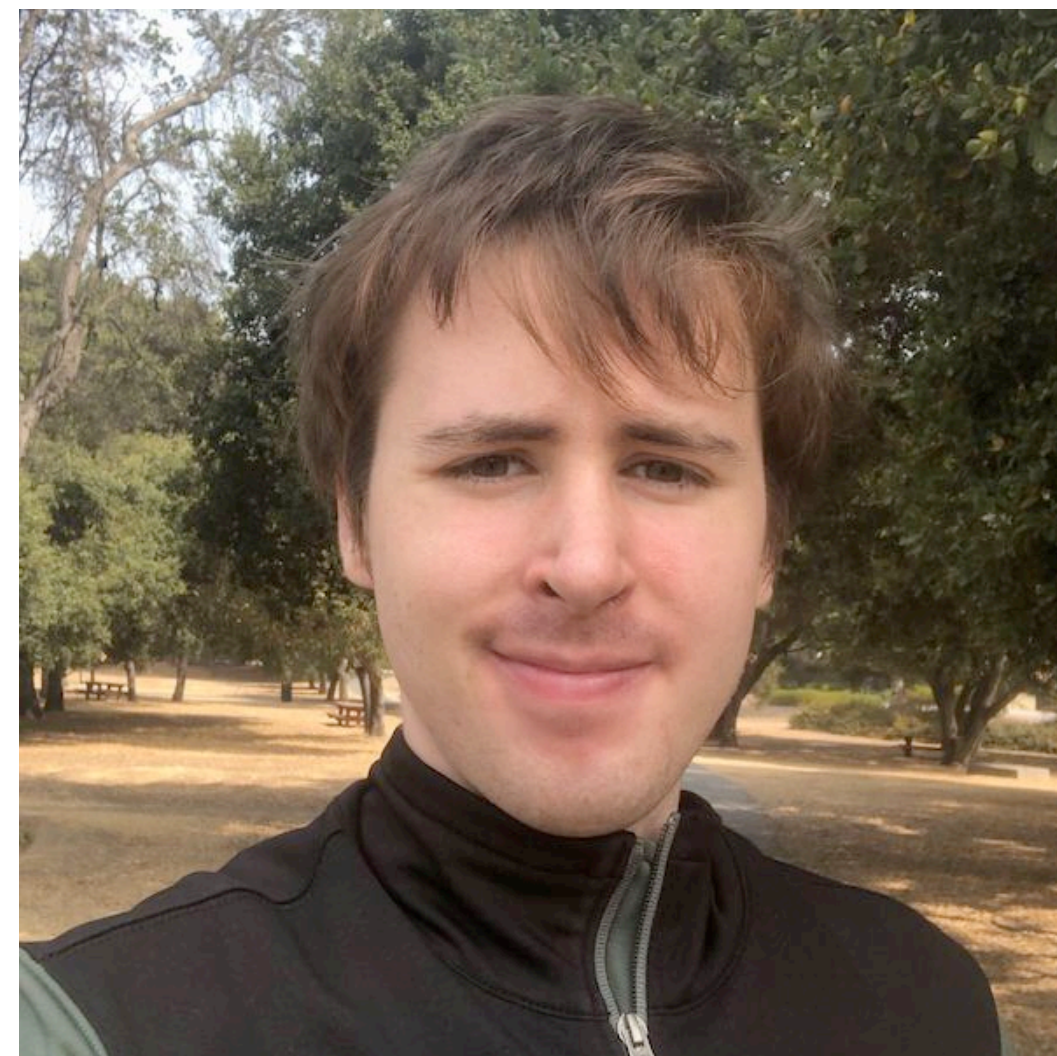
- I just graduated the coterm, focused on systems and security
- I was a community college graduate and transfer student
- I have two cats
- I love doing pottery, photography, and listening to music



Hello world! 🙌



Sophie



Patrick



Ayelet



Thea

Course website

- cs110.stanford.edu
- All course info, assignments, links, etc will be posted here

Lecture format

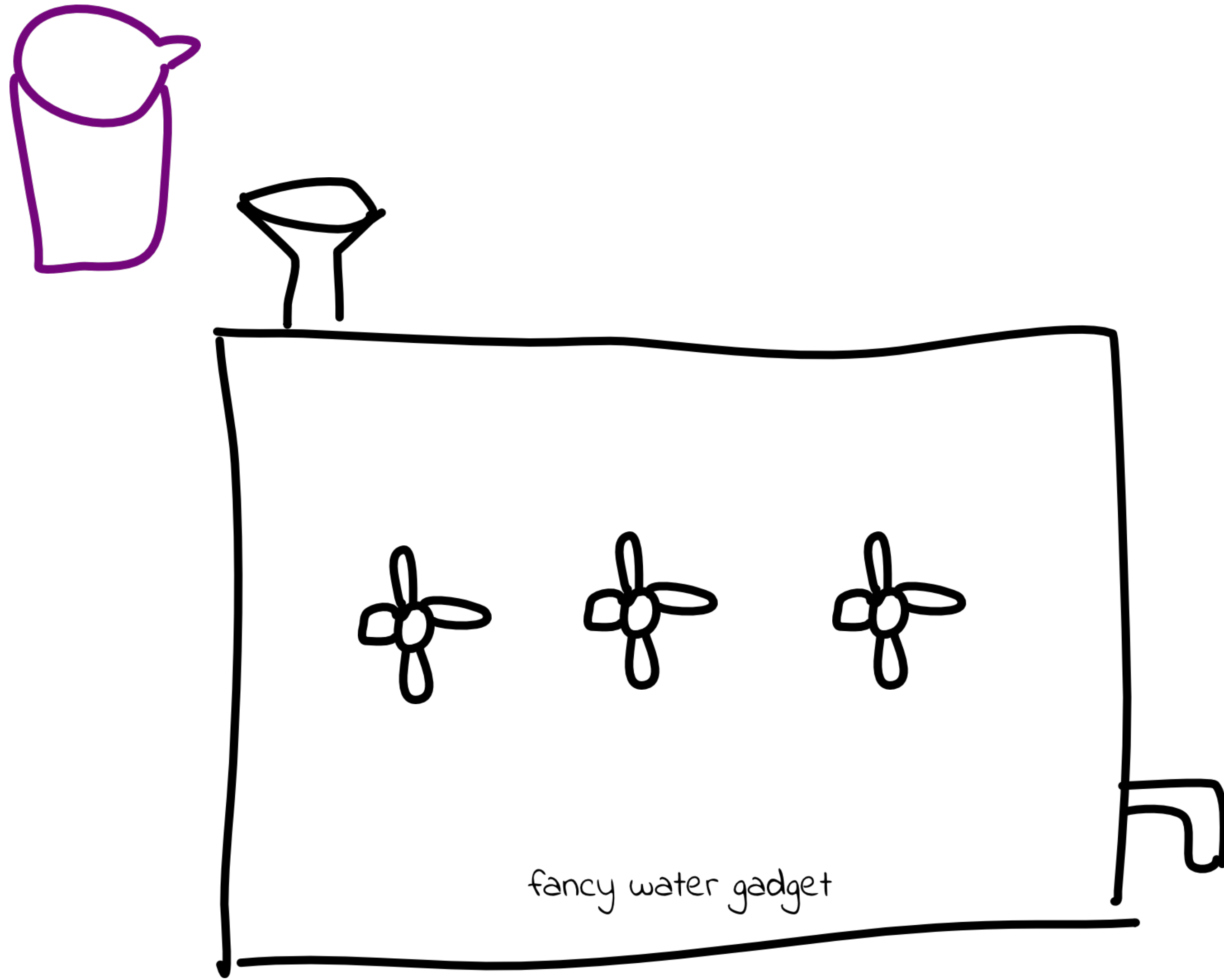
- Outside of lecture, you'll have plenty of practice with the “what” of systems programming; *in* lecture, we want to focus on the “why”
 - Going to be discussion-oriented as much as possible
- Lecture is synchronous — please come ready to participate!!
 - Pretty please :)
 - I'll come up with some creative incentives, e.g. I'll [make you a mug](#)
- Lecture is M/W/F week 1, only M/W after that

Edstem and Slack

- We'll be using Ed for the Q&A forum
- We'll be using Slack as a social forum, as well as group chat for your discussion section
- Links on the course website!

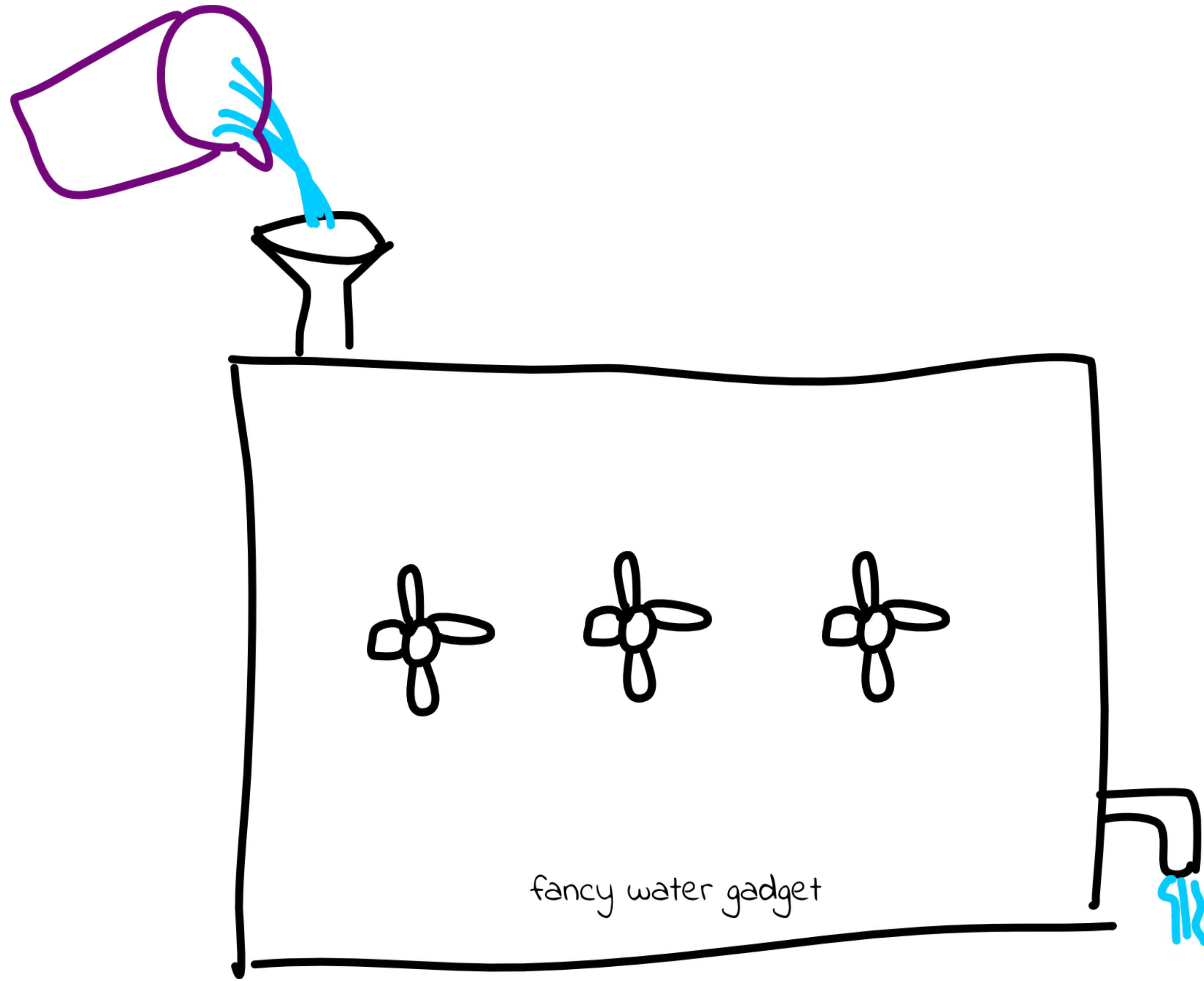
The art of debugging

The art of debugging



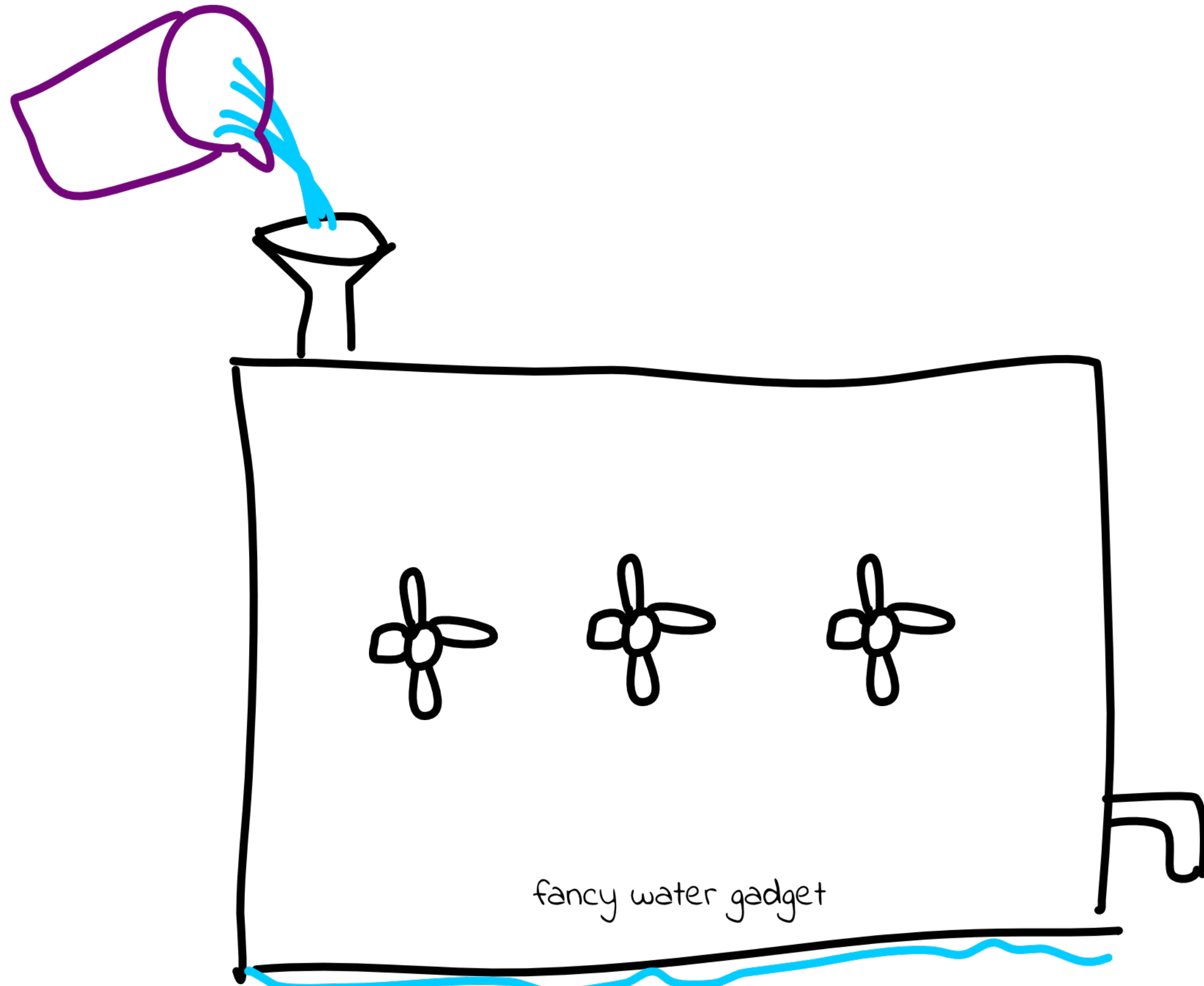
- Let's imagine we have a hypothetical "fancy water gadget"

The art of debugging



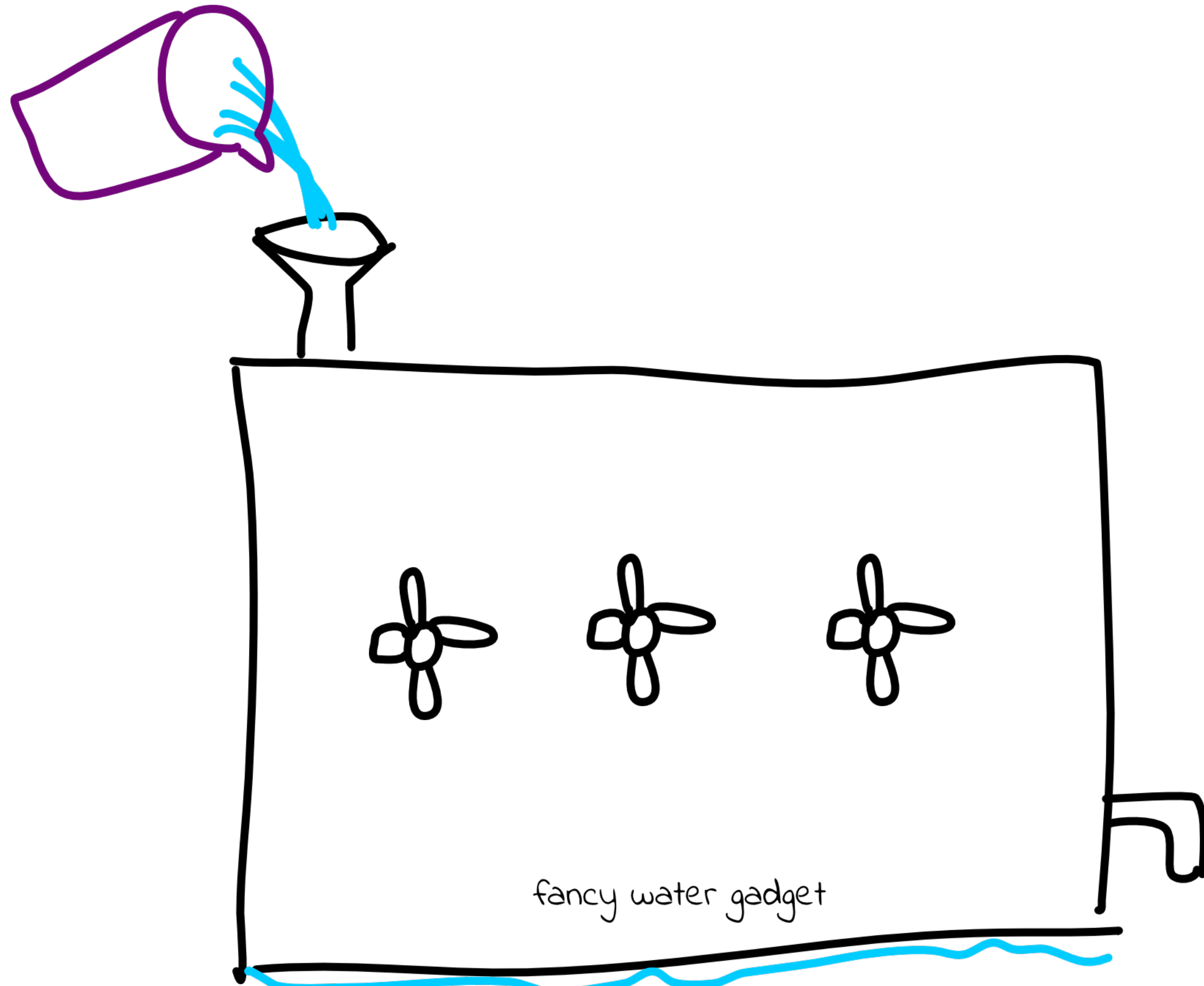
- Say we're supposed to pour water in on one end and get water out the other end

The art of debugging



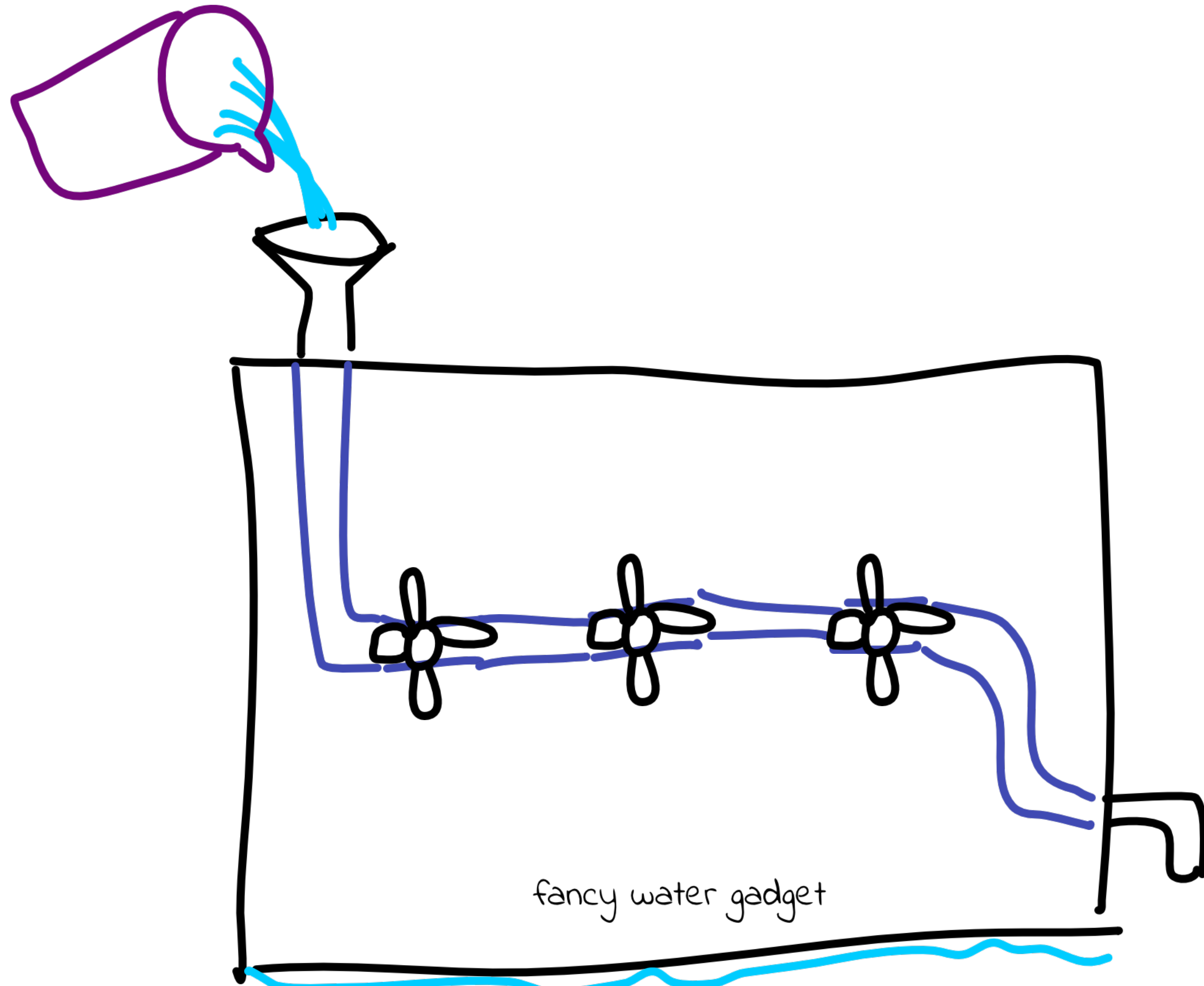
- Say we're supposed to pour water in on one end and get water out the other end
- But maybe this doesn't work, and water leaks instead
- What do you do?
- Beginner debugging strategies:
 - Stare at it until you figure it out
 - Mess with it until it works

The art of debugging



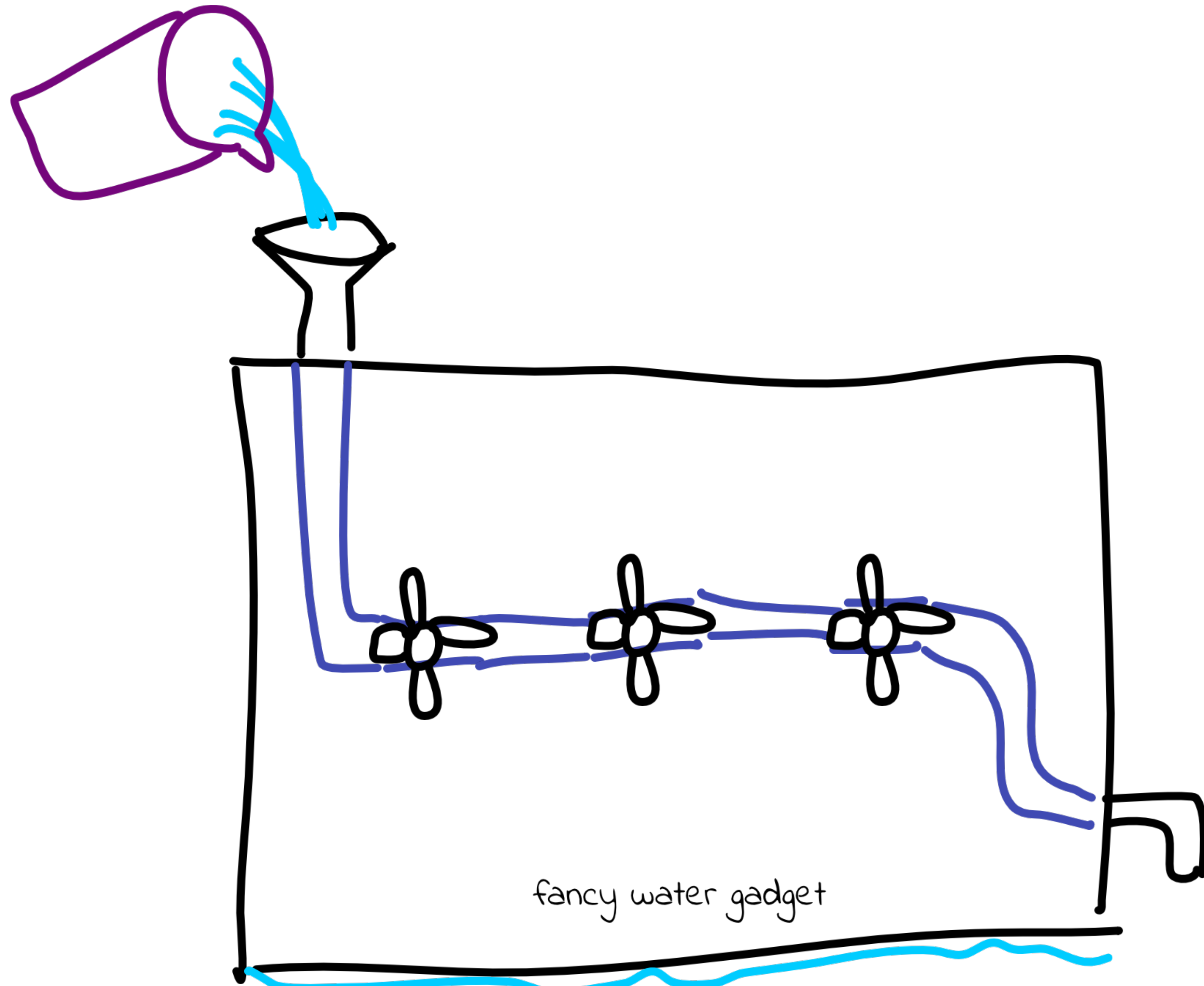
- There are two critical pieces to debugging effectively:
 - Develop a mental model of how things are *supposed* to work
 - Figure out what is happening *instead*

The art of debugging



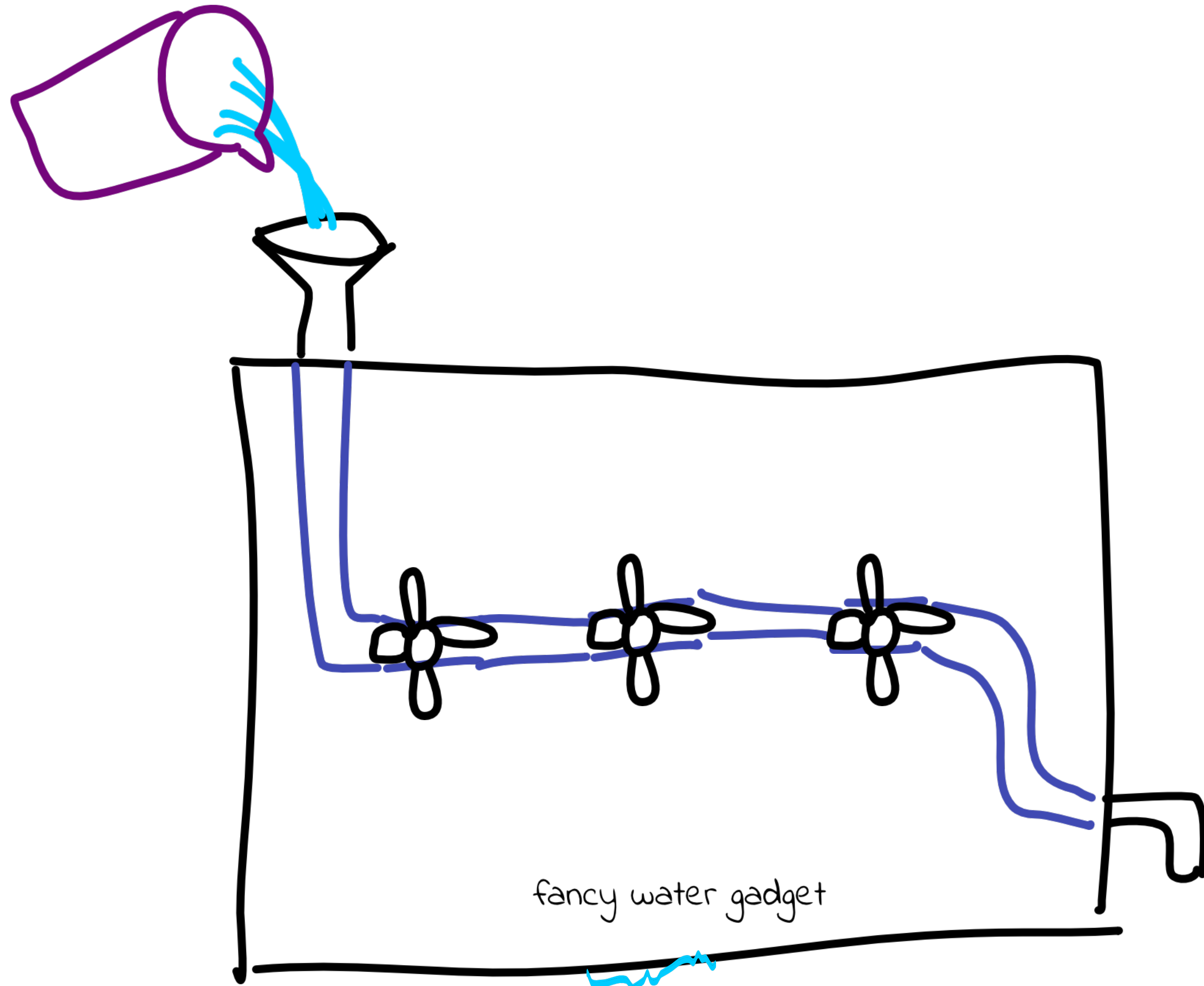
- #1: Develop a mental model of how things are *supposed* to work
 - Read documentation
 - Read Stack Overflow
 - Talk to people
 - Try experiments

The art of debugging



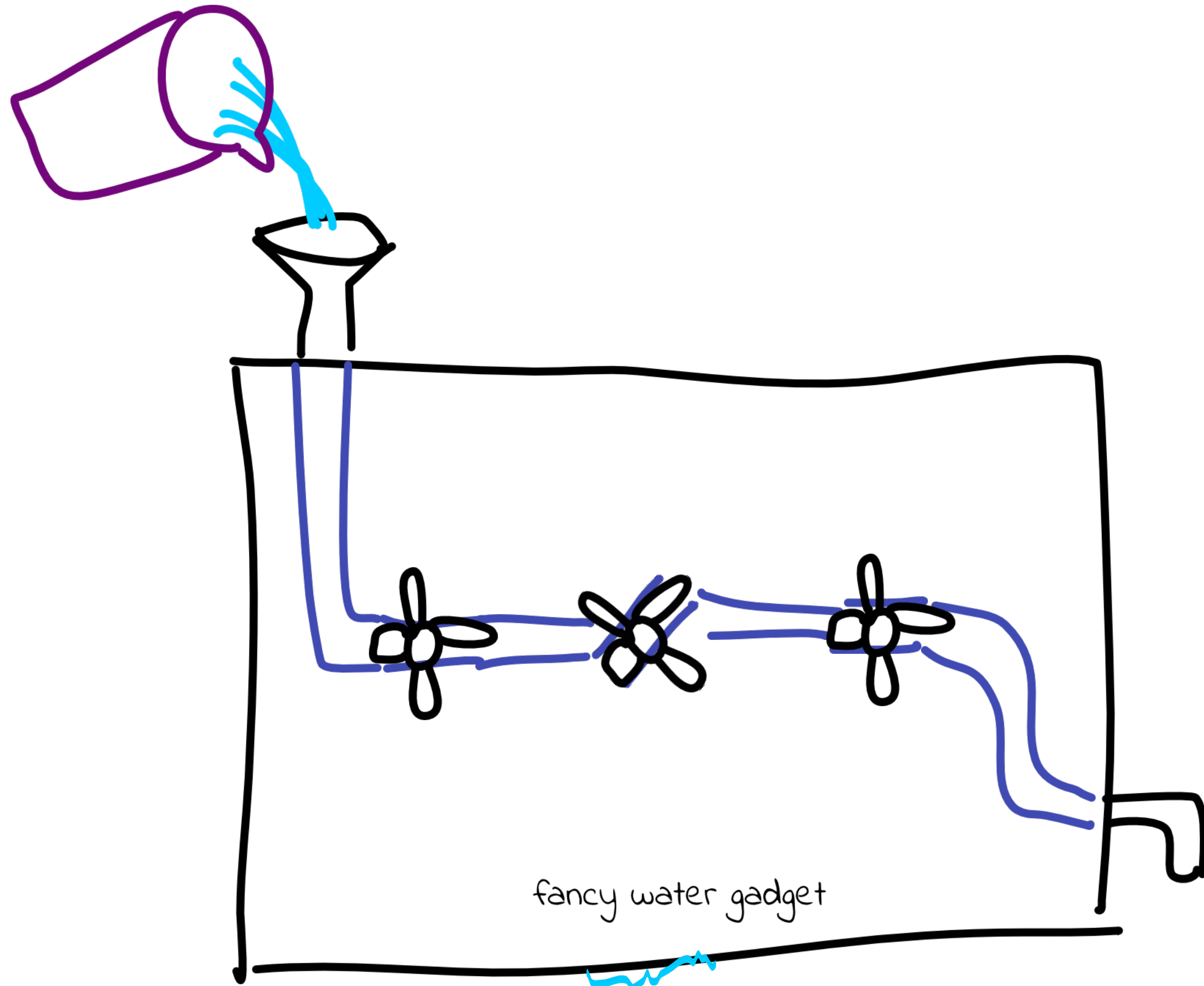
- #2: Figure out what is happening *instead*
 - Somehow need to get more visibility into the system
 - E.g. add print statements, use gdb, try other tools we'll discuss

The art of debugging



- #2: Figure out what is happening *instead*
 - Somehow need to get more visibility into the system
 - E.g. add print statements, use gdb, try other tools we'll discuss

The art of debugging



- #2: Figure out what is happening *instead*
 - Somehow need to get more visibility into the system
 - E.g. add print statements, use gdb, try other tools we'll discuss

Office hours

- Unlike previous classes, we will not look at code in office hours
 - One exception: if you're getting tripped up by syntax / compiler errors
- Our goal is to teach you how to debug your code, not to debug it for you
 - We will help make sure you have the right mental model of what is supposed to be happening
 - We will give you suggestions to get more visibility into what is actually happening
- We also want to encourage you to program carefully! Systems programming requires attention to detail

Grading

- Concept checks: 10%
- Discussion section participation: 10%
 - Sections don't start until week 2
- Programming assignments: 60%
- Self assessments: 20%

Late policy

- Life happens! Especially in these times
- We cap scores based on lateness:
 - On time: you get up to 100%!
 - Up to 24 hours late: 90% cap
 - 24-48 hours late: 70% cap
 - After 48 hours: please let us know what's up and how we can help
- I will grant extensions on a case-by-case basis. Please email me *in advance*

Honor code

- I will be relatively generous with deadlines...
- ...but I will not be very generous with honor code violations.
- There exist assignment solutions on the internet
 - We've been playing whack-a-mole taking these down, but it's not easy
- Lucky for us, these solutions have very subtle and unique mistakes
- 2 minute video from Brian Harvey on why you shouldn't cheat: <https://www.youtube.com/watch?v=hMloyp6NI4E>
 - Best case scenario, "you condemn yourself to a life of doing something you don't know how to do and don't like doing."
- Please do not be tempted to cheat! It is not worth it! We are here to support you

Filesystems

Considerations in filesystems

- A filesystem stores persistent data on some storage medium (e.g. hard drive)
- What are some things we might want from a filesystem?
 - Fast read/write performance
 - Low storage overhead (if I have a 1TB drive, I want to be able to store 1TB of files)
 - Resilient against data loss/corruption
 - Fancy features: encryption, compression, snapshots, etc.
 - Low complexity, easy to implement without bugs!!!

(live discussion)