CS111, Lecture 26

Ethics and Trust, continued

Learning Goals

- Reflect on aspects of trust, when we trust systems/others, and how we choose to trust systems/others
- Learn about examples of trust / isolation not being upheld in systems
- Discuss considerations for how to build trust into software we create
- Reflect on how technology affects trust

Trust + Operating Systems

- Properties of OSes (**immense scale**) provide a unique lens through which to examine how we **trust** software. All software trust comes back to OS.
- All software, especially OSes, can have a large impact on the people that use it, and they can put significant trust in that software.
- Examining ideas about trust can help us better consider how we might approach building trust into the software we build.

Plan For Today

- **Recap:** Who/what do we trust, and why?
- What do we do when trust is not upheld? (case study: Meltdown)
- How can we approach building trust into software?
- How does technology affect trust?

Plan For Today

- **Recap: Who/what do we trust, and why?**
- What do we do when trust is not upheld? (case study: Meltdown)
- How can we approach building trust into software?
- How does technology affect trust?

Recap: Trust So Far

What is trust?

- An unquestioning attitude – to stop questioning the dependability of something. Efficiency/safety tradeoff (trust is more efficient)
- Beneficial because it extends *agency:* our capacity to take actions that align with our goals

Ways to establish trust

- Assumption (weak, risky) – trust absent any clues to warrant it
- Inference (most powerful) – trust based on information
- Substitution ("Plan B") – trust by implementing system to partly replace need to trust something

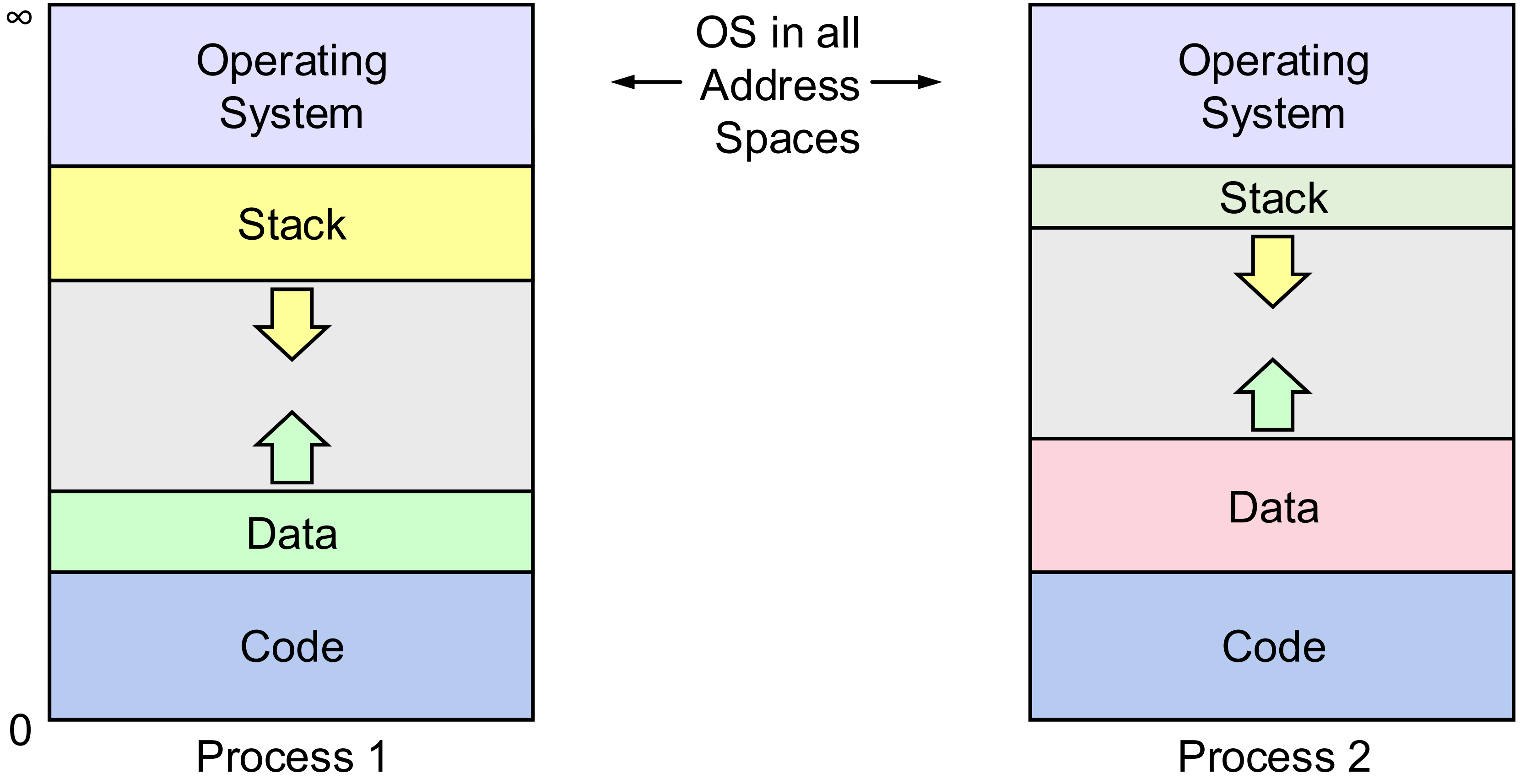
Trust is essential but risky

- Involves *intentions, dependence, vulnerability/risk*
- *Agential gullibility* – trusting more than is warranted

Plan For Today

- **Recap:** Who/what do we trust, and why?
- **What do we do when trust is not upheld? (case study: Meltdown)**
- How can we approach building trust into software?
- How does technology affect trust?

OS and User in Same Address Space



OS Execution

How does virtual memory work for the OS's needs?

OS has space in every process's virtual address space. Not a duplicate of OS; every virtual space could map to same physical memory.

Problem: don't want user program accessing OS pages.

Solution: new bit in page table that marks kernel-only pages. When in user mode, not accessible, but accessible when OS is running.

Meltdown

Meltdown is a **hardware vulnerability** publicly disclosed in 2018 that allows a program to access kernel-only pages. (https://meltdownattack.com)

> "Meltdown is a novel attack that allows overcoming memory isolation completely by providing a simple way for any user process to read the entire kernel memory of the machine it executes on, including all physical memory mapped in the kernel region."
> [meltdown paper]

Meltdown

Meltdown is a **hardware vulnerability** publicly disclosed in 2018 that allows a program to access kernel-only pages. (https://meltdownattack.com)

- fixes in later processors, patched in OSes

Relevant parties include:

- Hardware designers (e.g. Intel)
- OS designers (e.g. Microsoft Windows, Google Android, Apple iOS)
- App developers
- Users

Plan For Today

- **Recap:** Who/what do we trust, and why?
- What do we do when trust is not upheld? (case study: Meltdown)
- **How can we approach building trust into software?**
- How does technology affect trust?

Building In Trust

How can we approach building trust into software? Or, alternatively, what can we do as builders of software to increase the trust users have in our software?

- E.g. building in trust by substitution, for instance via
  - crash recovery mechanisms, or easy backup software (for instance **Time Machine** in macOS, or backing up an Android phone via Google account).
  - Easy data export if the software is deprecated
  - Etc.

Building In Trust

Considerations might include:

- Who are the stakeholders?
- How pervasive is it?
- What are the long-term intentions?

Identifying each of these helps focus efforts to build trust into software we build.

Stakeholders

Key Question: *who will be interacting with or affected by the system?*

Why is this important? *Identifying stakeholders allows us to focus our trust efforts.*

Direct stakeholders: directly interact with the system

Indirect stakeholders: affected by the system without directly using it

<u>Example:</u> medical office use device

- **Direct stakeholders:** medical professional operating the device, service technician maintaining device
- **Indirect stakeholders**: patients

Pervasiveness

Key Question: *what is the use case, and how widespread is it?*

Why is this important? *pervasiveness may influence how we approach building in trust – e.g. maybe critical infrastructure software would make different tradeoffs (e.g., cost, performance, reliability) vs. personal software.*

- How widespread is the use?
- What is the use case? (personal, recreation, critical infrastructure)
- Considerations about crossing national boundaries (different rules, customs, infrastructure)
- Considerations about cultural and political implications

Time

Key Question: *what is the duration/longevity of the software? How do we manage end of support?*

Why is this important? *Software timescale changes how we consider building in trust – e.g. will users be relying on it for a long time?*

- Support duration (long-term support) - assign2
- Obsolescence: how do we manage end of support?
  - **Example:** long-term support for operating systems (assign2)
  - **Example:** software updates
  - **Example:** online services (e.g. games with online components)

Time

Key Question: *what is the duration/longevity of the software? How do we manage end of support?*

Why is this important? *Software timescale changes how we consider building in trust – e.g. will users be relying on it for a long time?*

- Other scenarios where product/support may not be guaranteed forever
  - **Example:** subscription content services
  - **Example:** company going out of business, no longer maintains/supports product
- What does halting use look like?
  - **Example:** Threads app initially not supporting account deletion without deleting Instagram account

Discussion Question

You are creating software to track personal medical information (e.g. doctor’s visits, vitals, other health information, etc.).

- Concept: doctors could input information, and patients could view and add their own information.

Considering **stakeholders**, **pervasiveness** and **time**, what are some ways that we could build trust into the product? Or alternatively, what features would increase trust for you as a user if you were using this product?

Once you have discussed, respond on PollEv: pollev.com/cs111

You are creating software to track personal medical information. Considering stakeholders, pervasiveness and time, what are some ways that we could build trust into the product?

Nobody has responded yet.

Hang tight! Responses are coming in.

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Plan For Today

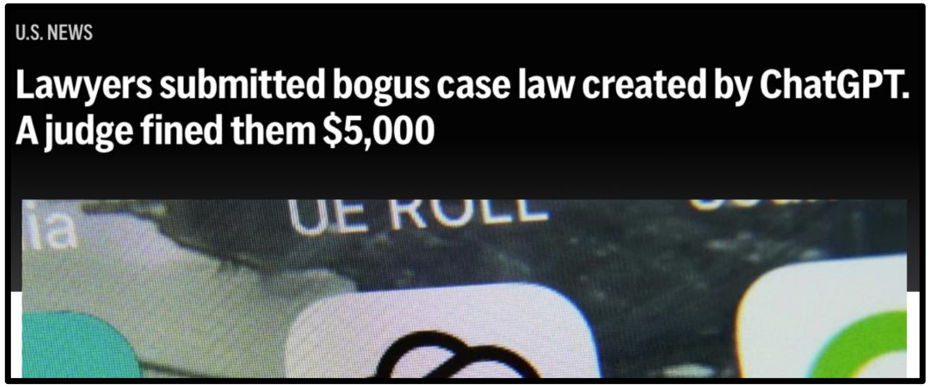
- **Recap:** Who/what do we trust, and why?
- What do we do when trust is not upheld? (case study: Meltdown)
- How can we approach building trust into software?
- **How does technology affect trust?**

How does technology affect trust?

Sometimes technology can lead to agential gullibility.

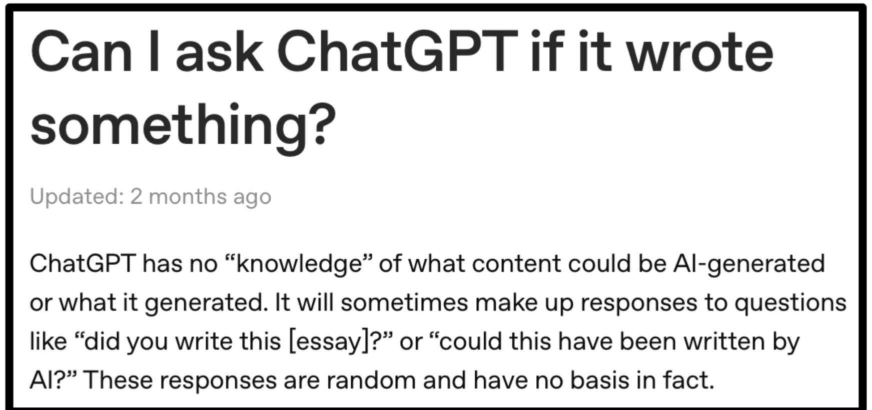
Example: **ChatGPT**

- Generative AI tools can produce useful and insightful information
- ChatGPT presentation causes people to infer trust:
    - Authoritative, with explanations (Bansal et al. 2021)
    - Lots of concrete "facts" (Bower et al. 2024)
- But, ChatGPT hallucinates; no reason to trust!
    - Various examples of agential gullibility: submitting fake court case info, asking if it wrote something



U.S. NEWS

Lawyers submitted bogus case law created by ChatGPT. A judge fined them $5,000



Can I ask ChatGPT if it wrote something?

Updated: 2 months ago

ChatGPT has no "knowledge" of what content could be AI-generated or what it generated. It will sometimes make up responses to questions like "did you write this [essay]?" or "could this have been written by AI?" These responses are random and have no basis in fact.

How does technology affect trust?

Sometimes technology can lead to agential gullibility.

Example: **ChatGPT**

- Generative AI tools can produce useful and insightful information
- ChatGPT presentation causes people to infer trust:
  - Authoritative, with explanations (Bansal et al. 2021)
  - Lots of concrete “facts” (Bower et al. 2024)
- But, ChatGPT hallucinates; no reason to trust!
  - Various examples of agential gullibility: submitting fake court case info, asking if it wrote something
- Embedding ChatGPT in other apps obscures origin of information

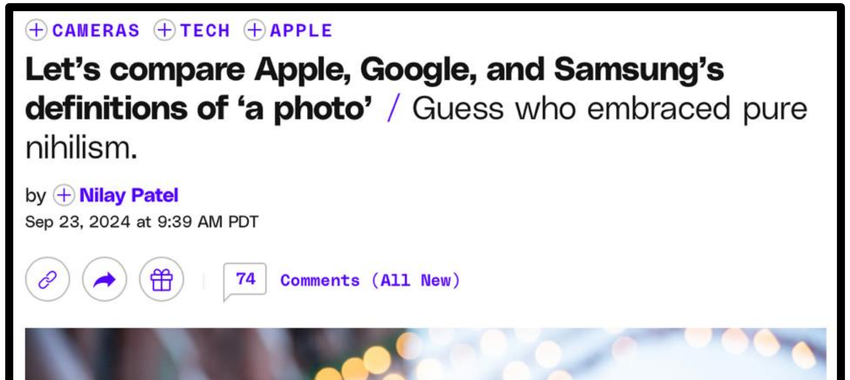
Takeaways: treat output as hypotheses to consider, validate results (use substitution)

How does technology affect trust?

Sometimes technology can require us to re-evaluate what we trust.

Example: **AI-generated/edited imagery**

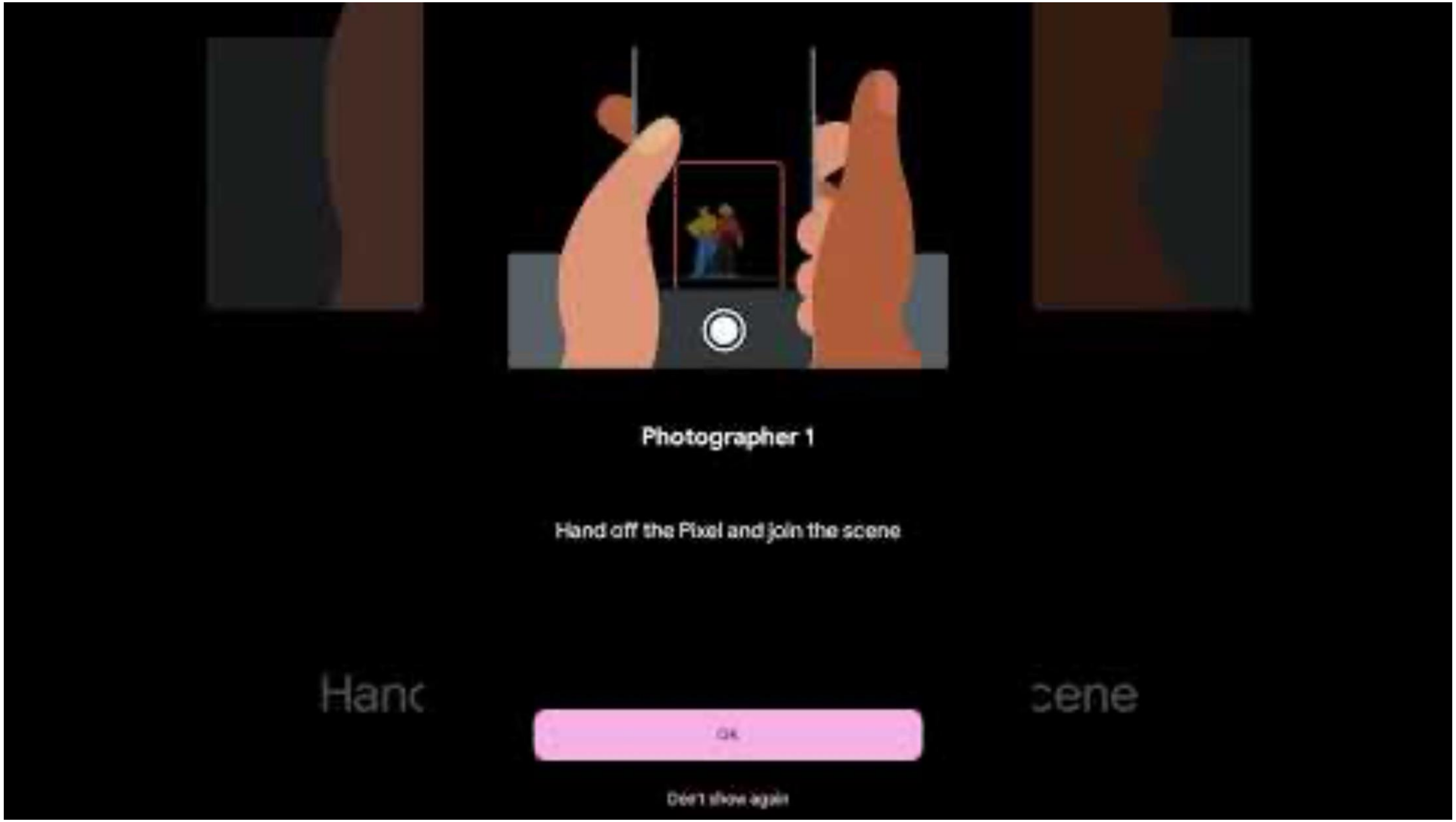
- Historically: harder to fabricate convincing photos, videos, audio (though not new – e.g. Photoshop)
- People inferred trust (for good reason)
- New technology enables alterations or completely fake photos
  - What is a photo? The Verge. E.g. “magic eraser” photo editing
  - Google “Add Me” feature for inserting people into photos: video demo



Let’s compare Apple, Google, and Samsung’s definitions of ‘a photo’ / Guess who embraced pure nihilism.

by Nilay Patel
Sep 23, 2024 at 9:39 AM PDT

Google “Add Me”



Photographer 1

Hand off the Pixel and join the scene

How does technology affect trust?

Sometimes technology can require us to re-evaluate what we trust.

Example: **AI-generated/edited imagery**

- Historically: harder to fabricate convincing photos, videos, audio (though not new – e.g. Photoshop)
- People inferred trust (for good reason)
- New technology enables alterations or completely fake photos
  - What is a photo? The Verge. E.g. “magic eraser” photo editing
  - Google “Add Me” feature for inserting people into photos: video demo
- Technology can also contribute to *increasing* trust – e.g. work on SynthID for identifying AI-generated content.

Takeaways: must unlearn trust in photos, videos, audio – do not trust without validation.

Bailey Surfing? 🐶 (credit: ChatGPT)

Discussion Question

What are some other examples of technology leading to agential gullibility, or requiring us to re-evaluate what we trust?

Key Takeaways

Trust is often required, powerful, and dangerous. We must consider:

- what we trust and why – extending agency, “unquestioning attitude”
  - Trust by assumption, inference, substitution
- what may happen if trust is not upheld (case study: Meltdown)
- How we can design trust into products we create (e.g. structures that enable us to substitute trust).
  - Considerations include stakeholders, pervasiveness, and time
- How technology impacts trust. E.g. ChatGPT, AI imagery

Help evaluate the Embedded Ethics Program!

https://tinyurl.com/embeddedethicsw26

10-15 minute survey, taking it (or not) won't impact your grade in the class in any way. The teaching team won't know who participates.

Option to provide your **Stanford email address** to receive a **$10 gift card**, **up to the first 800 participants**. Compensation once per quarter (SUNet login required).

Questions? Email embeddedethics@stanford.edu

Stanford | Embedded Ethics

Plan For Today

- **Recap:** Who/what do we trust, and why?
- What do we do when trust is not upheld? (case study: Meltdown)
- How can we approach building trust into software?
- How does technology affect trust?

Lecture 26 takeaway: Trust is often required, powerful, and dangerous. We must consider what we trust and why, what may happen if trust is not upheld, how we can design trust into products we create, and how technology impacts trust.