Reflections on F/OSS Design + Closed PDK:

If You Can't Trust the Transistors, Why Bother With Anything Else?

bunnie (@bunniestudios / twitter)

Silicon Salon - 2023

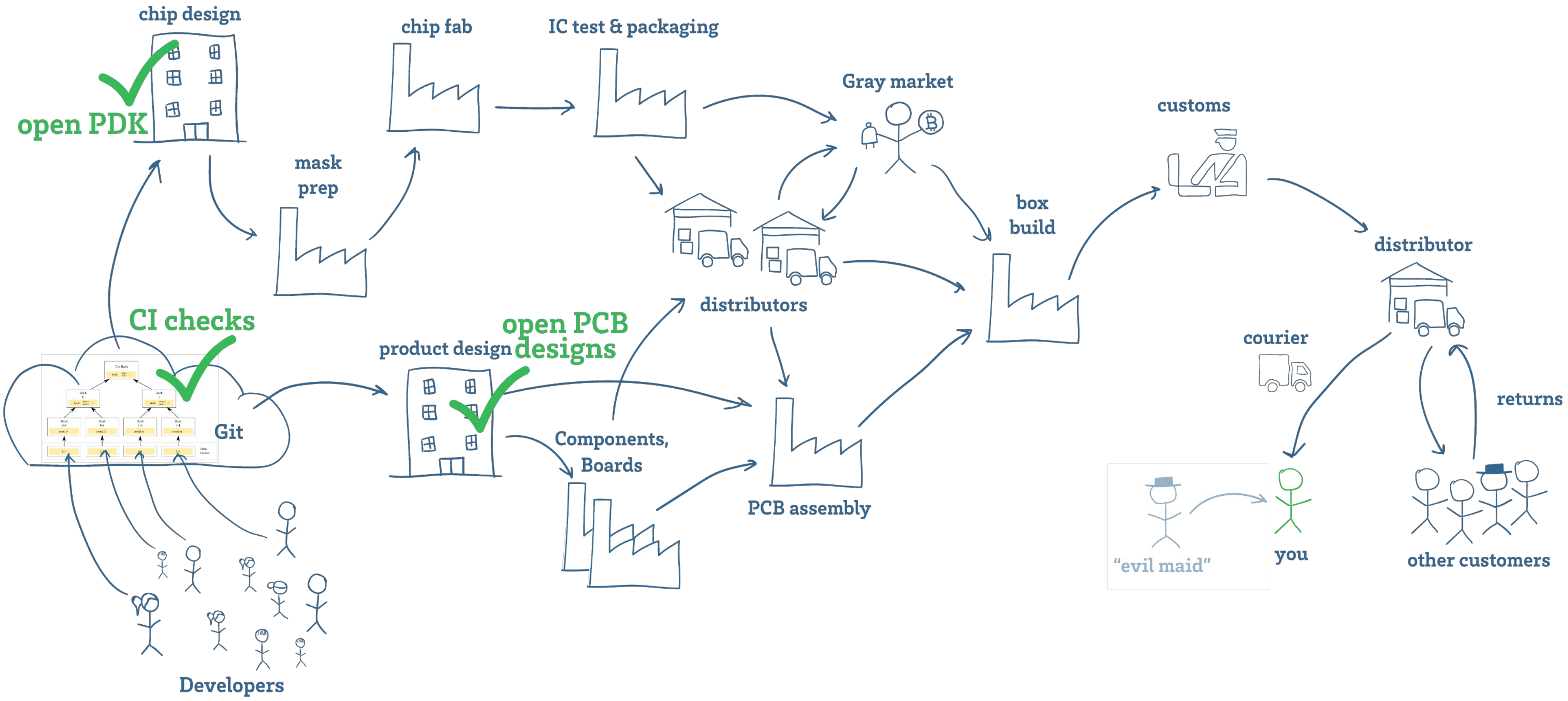
So You Care about Security, and You Want to Trust your Hardware.

- Kerckhoffs's principle: avoid security through obscurity
  - So, Open all the things!
    - Protocols/Apps
    - Kernel
    - Firmware/bootloaders
    - Circuit boards
    - Chips
    - RTL
    - PDK
    - Masks
    - Chip fabs...

Alternatively Stated:
What If You're Trapped in a Simulation?

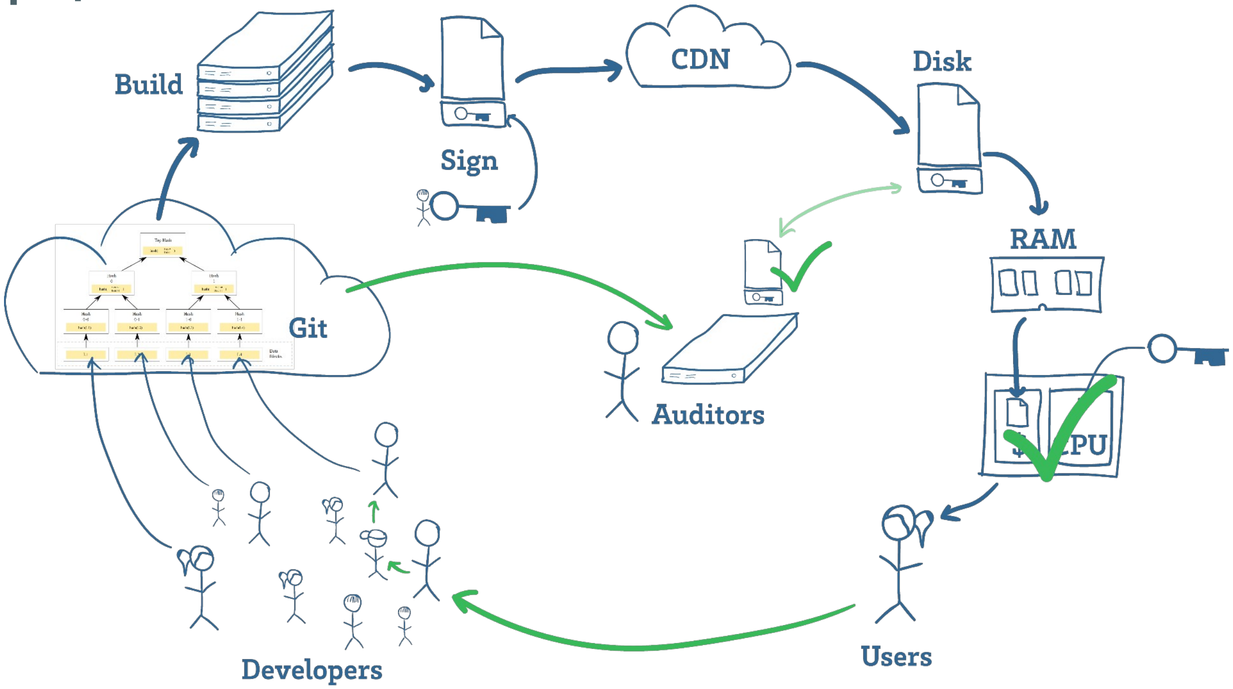
- If your BIOS is rooted, does it matter that your kernel is trusted?
- If your motherboard has a JTAG implant, does it matter that your BIOS is signed?
- If your CPU has patched microcode, does it matter that your motherboard is trusted?
- If your CPU microcode is signed, does it matter if the chip design is back-doored?

The Turtles Stop Here: Open PDK?



In Hardware, Checked Designs Does Not Mean Checked Devices

- Trust cannot be transfered from design to device via cloud
- There is no "hash function" + "digital signature" for hardware
- (At least not yet)



So, I am Worried about Backdoors in Chips: Inspect All the Chips, Down to the Transistor?

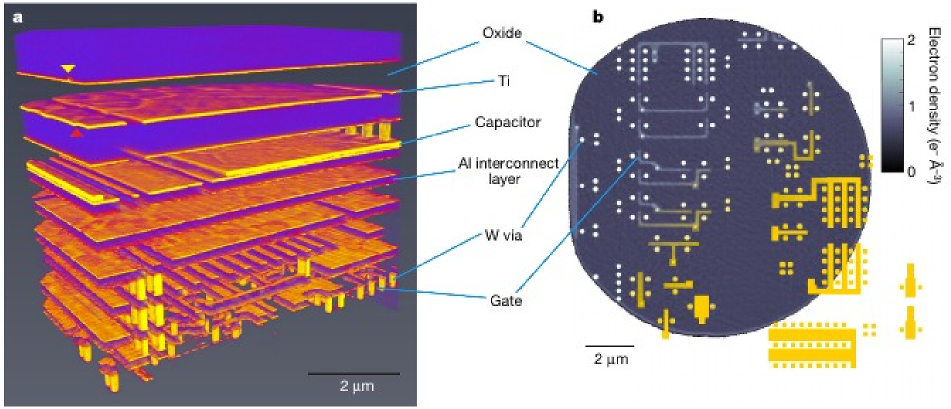


Figure 2 | PXCT of detector ASIC chip. a, 3D rendering of the PCXT tomogram with identified elements. The yellow triangle indicates a manufacturing fault in the Ti layer. The Al layer in the region of the red triangle shows variances in thickness causing a waviness of the Ti layer on top. Via, through-layer connector. **b**, Axial section across the second lowest layer, which contains the transistor gates; the grey scale (top right) represents electron density (in e^{-} Å^{-3}). The corresponding layer from the design file is shown as the partial overlay in yellow.

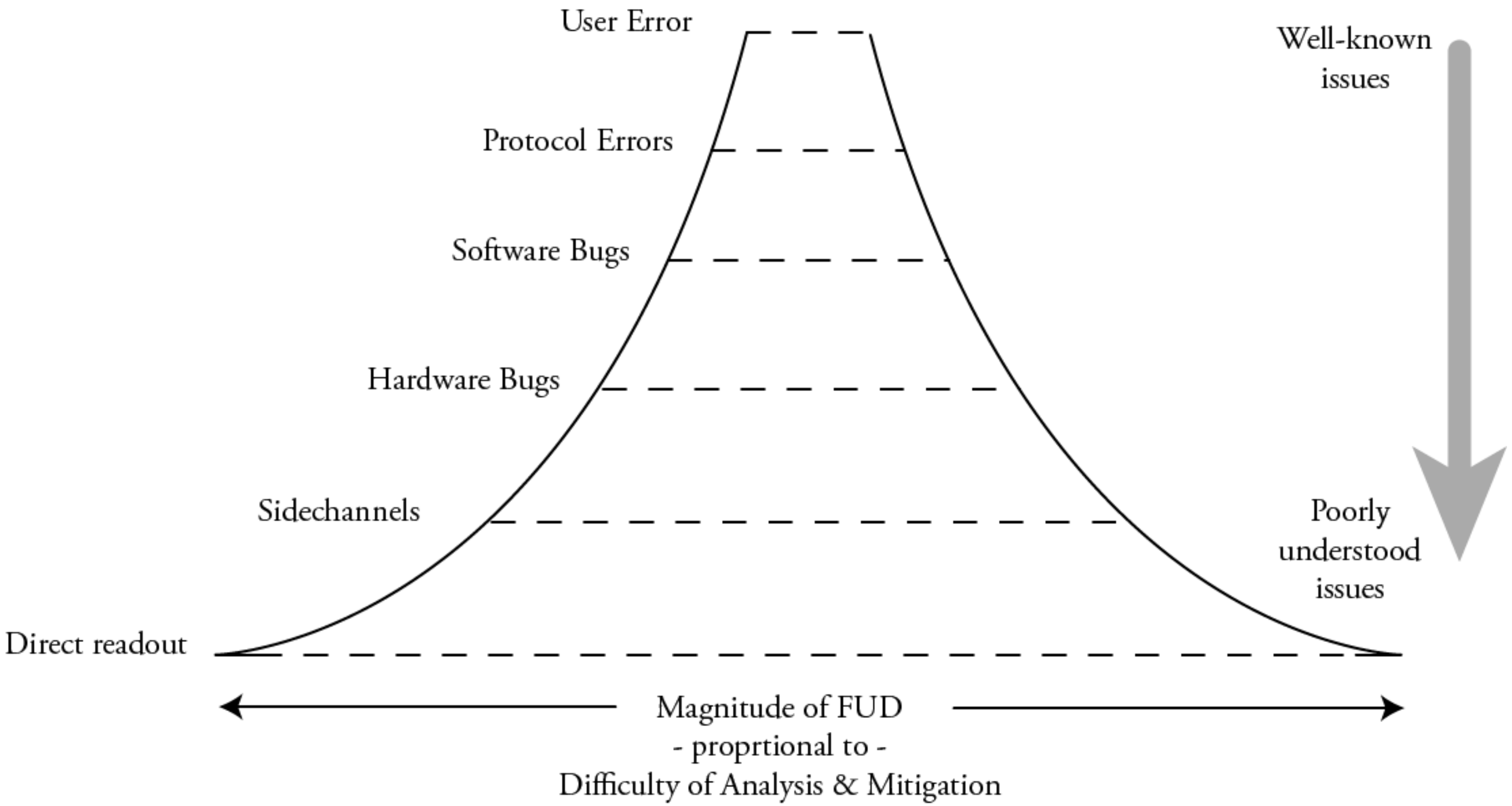
I Have Bad News

- There are no "silver bullets" in hardware security
  - Formally verification has no essential link with security
  - Open source has no essential link with trustability
  - Physical inspection has limits
  - Yesterday's inspection does not ward off today's "evil maid"
  - Trusted fabs are meaningless with untrusted couriers
  - Audits cost money
  - Certifications are a business, not a public service

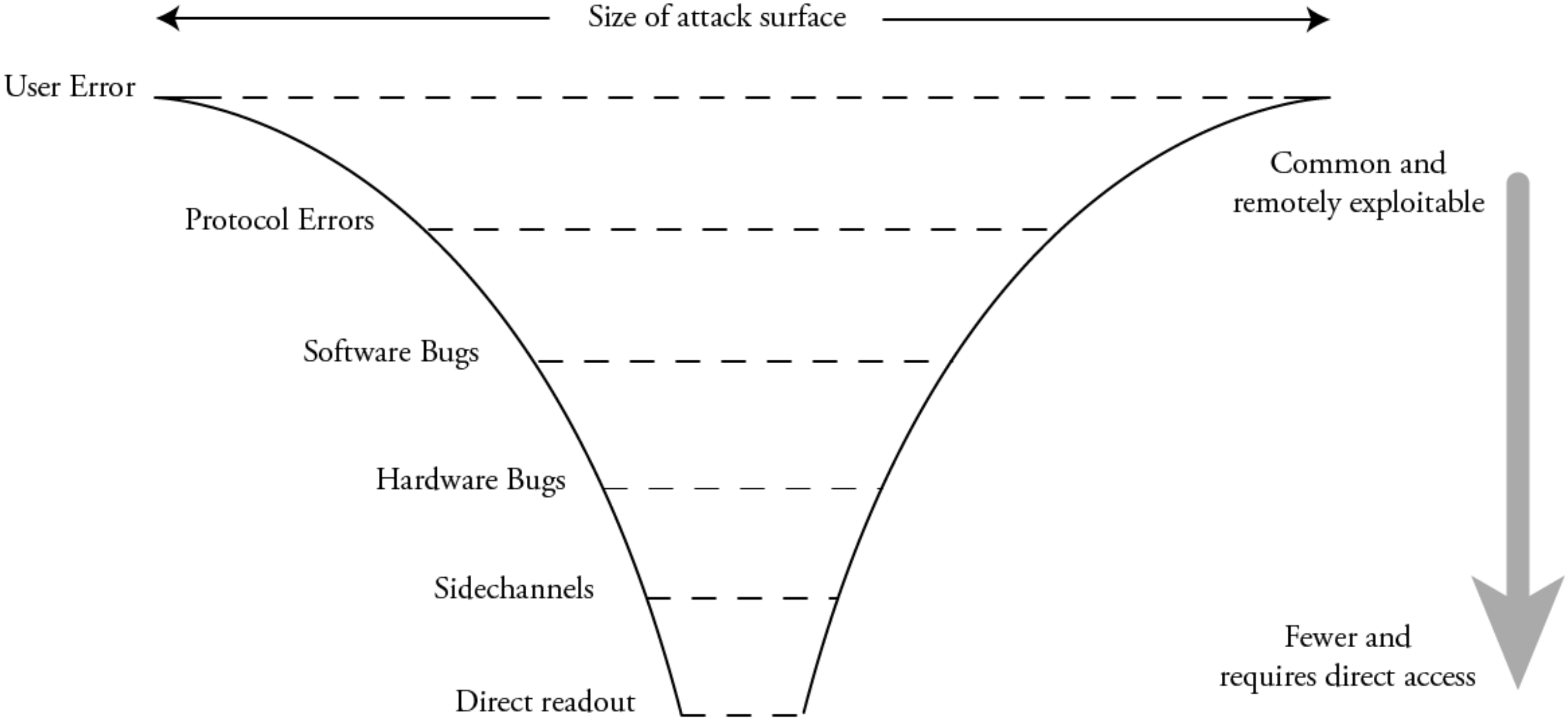
Hardware Security is a Cost-Benefit Tradeoff

- How much does it cost to break the security?
- How much do you lose if the security is broken?
- Accurately asesssing these costs is fundamental!

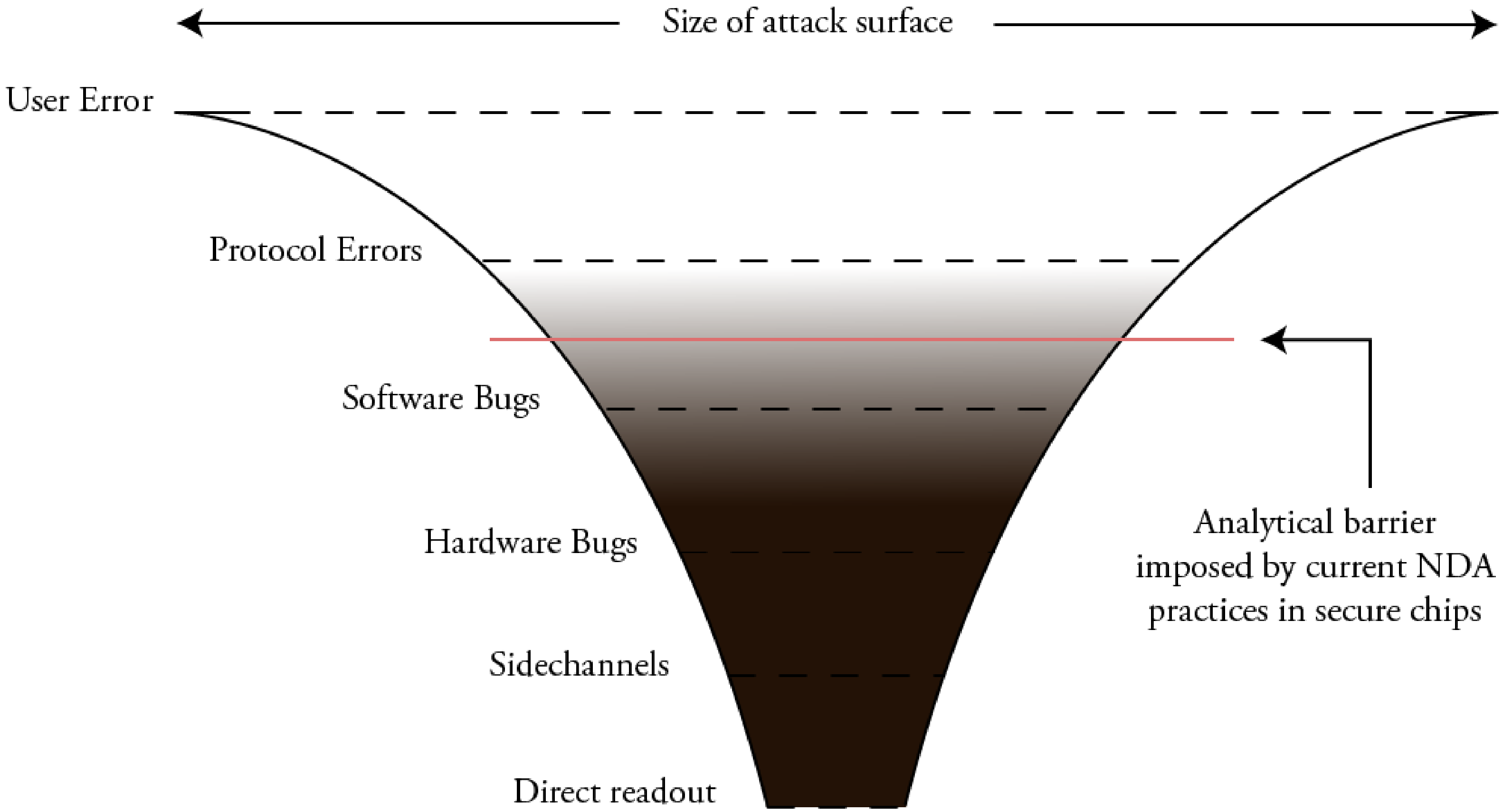
Why Cost Assesment is Hard:
Fear is Proportional to Uncertainty



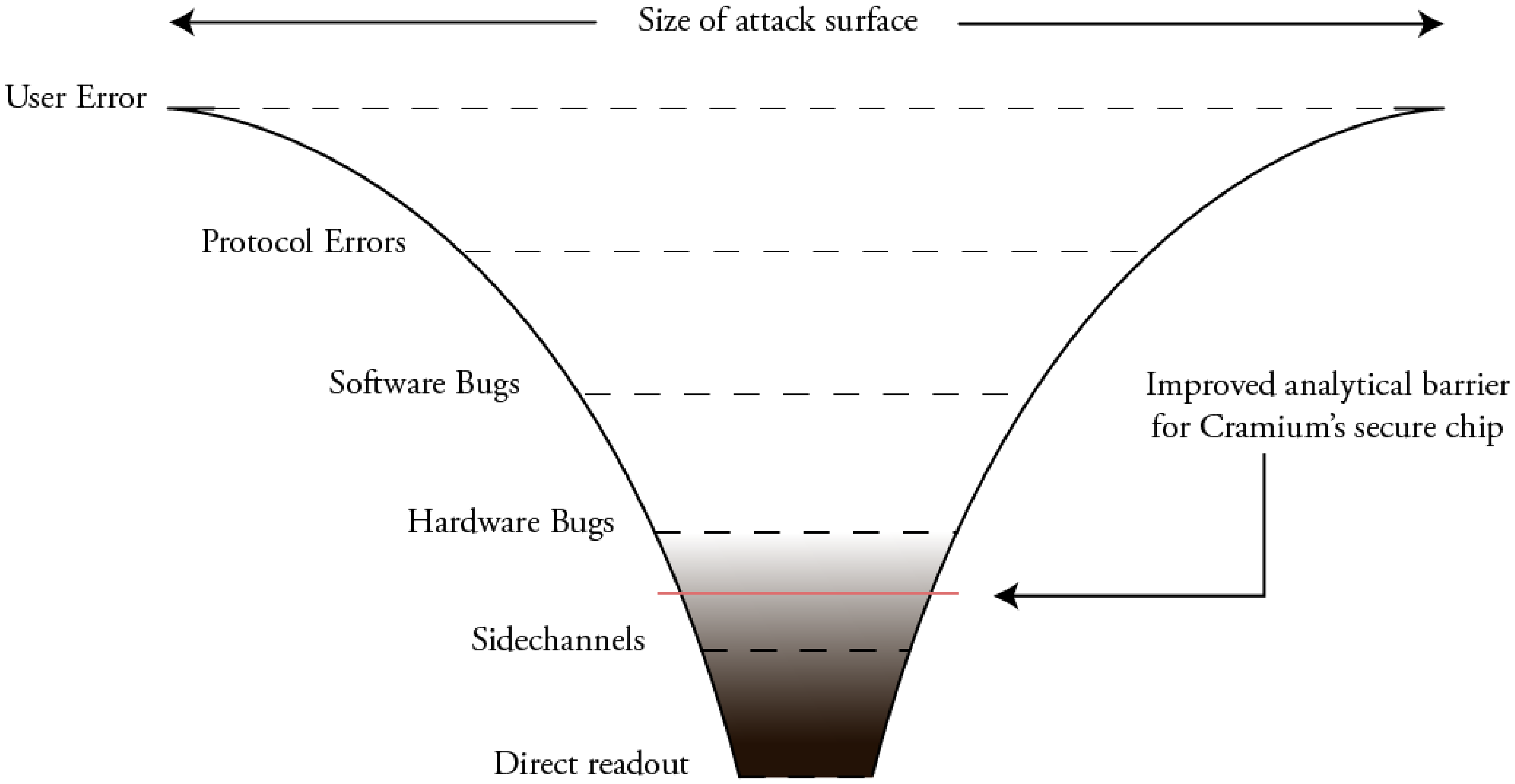
A Possibly More Accurate View of Attack Surface Size



The Impact of Closed Hardware Extends Beyond the Surface of Hardware



The Effect of Moving the Analytical Barrier Down the Stack

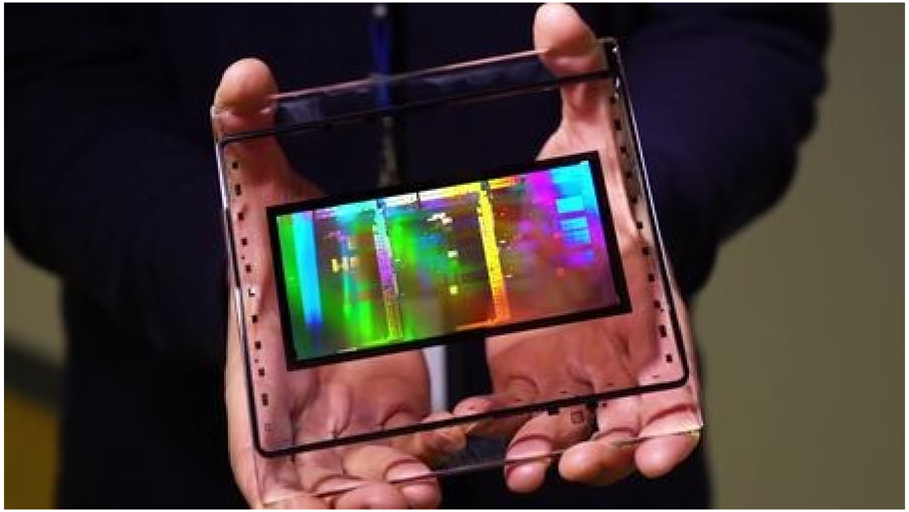


RTL-Level F/OSS Design, on a Closed PDK Pros & Cons

- Pros
  - Reduction of software bugs assisted by analysis of hardware design
  - Faster & analytical patching of hardware bugs
  - Bug or backdoor? Now we can know
  - Some improvement in physical inspectability (gross morphology is constrained)
- Cons
  - Can't be sure the transistors match the RTL
  - No improvement in analytical difficulty for sidechannel/direct readout vectors
  - Does not improve transistor-level inspection
  - Still standing on turtles

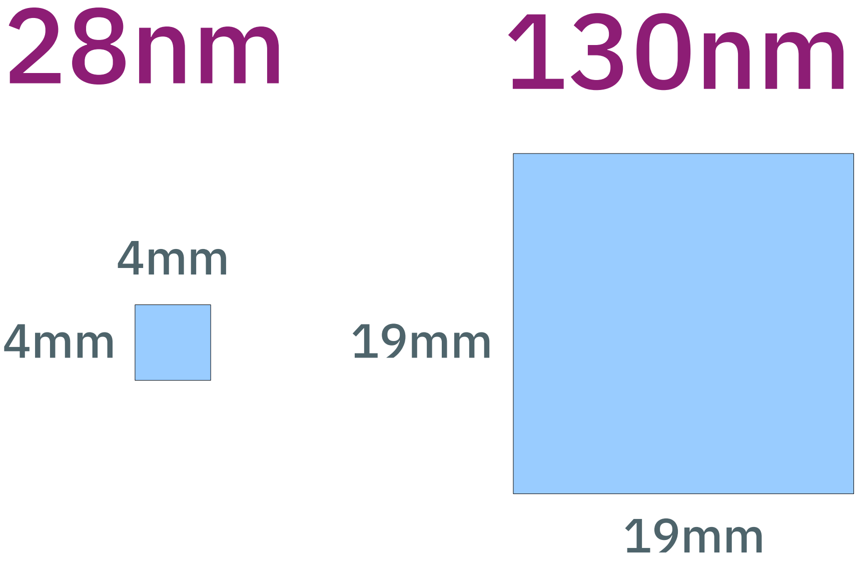
If All Things Were Equal:
Of Course, a Fully Open PDK Is Better

- The basic strawman goes:
  - Security is important
  - Reticles are huge
  - Just fab your security chip on 130/180nm open PDK processes, and use a full reticle



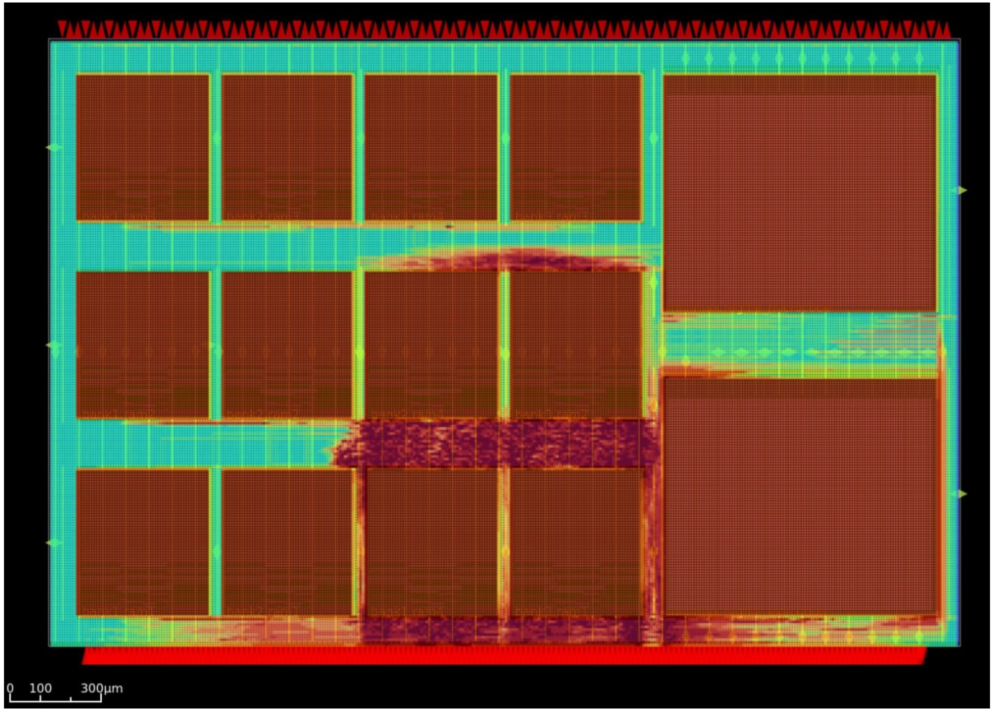
Problem #1: Physics, Form Factor, Economics

- Assume:
  - Same RAM/ROM capacity
  - Same microarchitecture
- Cost difference
  - 20x: $5 chip -> $100 chip
- Speed or power difference
  - 5-10x(?) power/speed scaling differential
- Form factor
  - A 19x19mm chip can't fit in a smartcard



Problem #2: Not all PDKs are Equal

- The current 130/180nm PDKs come with limitations:
  - Poor SRAM support
  - Few analog blocks
  - Effort, time & validation still to be done to optimize PDK for prime-time



(credit: Sean Xobs Cross)

2.92x3.52mm GF180
8k RAM (left)
Register files (right)

Problem #3: Opportunity Costs

- Outside of the security research field:
  - Security is a barrier to adoption
  - Hard to up-sell as a feature
- Security tends to settle around standards
  - e.g. "Don't roll your own"
  - First-movers have the ability to set de-facto standards around closed-source/proprietary primitives
    - e.g. ARM microarch + MPU
    - Microarchitectural lock-in is real: x86 vs the world

So Which Is Better?

- Bottom-up approach:
  - PDK
  - RTL
  - API
  - OS

- Top-down approach:
  - OS
  - API
  - RTL
  - PDK

Porque No Los Dos?

Q&A

@bunniestudios

@bunnie@treehouse.systems