

A practical reading list for AI’s limits—and the systems that contain failure

Recommended Reading from Tech Founders

2026-09-21

A practical reading list for AI’s limits—and the systems that contain failure

By Recommended Reading from Tech Founders • September 21, 2026

A curated set of organic recommendations spanning secure-systems research, an AI-research debate, a draft AI-use policy, and a few personal picks from product and tech leaders.

If you read one cluster, make it Martin Casado’s pairing of older systems research with a current AI-cyber concern. He argues that AI cyber capabilities should force us to build secure systems “all the way down” and names HYDRA and Asbestos as two of his favorites. [1] The payoff is specificity: one paper makes protection and extensibility a kernel-design problem; the other turns isolation and information flow into explicit mechanisms. [2, 3]

Start with the systems-security pair

HYDRA: the kernel of a multiprocessor operating system

- **Content type / creator:** Research paper by W. Wulf et al., published in *Communications of the ACM* in 1974. **Link:** ACM record. [2]
- **Recommended by:** Martin Casado, as one of his two favorite works on building secure systems. [1]
- **Key takeaway:** HYDRA treats physical and virtual resources as “objects,” then supplies mechanisms for creating types, defining operations, sharing, and protecting references. The paper presents that as a basis both for extending the system and for creating highly secure systems. [2]
- **Why it matters:** It gives readers a concrete vocabulary for asking where an AI-enabled system’s resources live, who can operate on them, and how those permissions are enforced.

Labels and event processes in the asbestos operating system

- **Content type / creator:** Research paper by Petros Efstathopoulos et al., published in *ACM SIGOPS Operating Systems Review* in 2005. **Link:** ACM record. [3]
- **Recommended by:** Martin Casado, alongside HYDRA. The ACM record supplies the corrected title; his post spells it “Absestos.” [1, 3]
- **Key takeaway:** Asbestos uses kernel-enforced labels and isolation to control inter-process communication and system-wide information flow. Its event-process abstraction lets one process act for multiple users without leaking one user’s data to another; the abstract reports about 1.5 memory pages per user for a labelled web server. [3]
- **Why it matters:** This is a useful counterweight to abstract AI-security warnings: it shows what containment can look like at the operating-system boundary, including an explicit cost model.

The best current AI-research debate

The Hinton–LeCun exchange on AI research and safety

- **Content type / creators:** X thread involving Geoffrey Hinton and Yann LeCun, as described by Chamath Palihapitiya. **Link:** LeCun thread.
- **Recommended by:** Chamath calls it “worth reading” and describes it as a 2023 back-and-forth about the tension between AI research and safety. He highlights LeCun’s warning that doomerism could be used to lock up open research, open-source code, and open-access models, and calls LeCun’s view that current state-of-the-art AI is still “just software” but incomplete prescient. [4]
- **Key takeaway:** The linked LeCun argument turns that dispute into technical tests: current reasoning relies on non-autoregressive search but in a limited token space; self-improvement works mainly where outputs can be scored without human intervention; and current systems still lack rapid adaptation to previously unknown situations. [5]
- **Why it matters:** It is a way to interrogate both safety claims and capability claims with concrete questions about search, learning, embodiment, and transfer—not just with “doomer” or “booster” labels.

A policy artifact for AI-assisted work

Proposed KDE LLM guidelines

- **Content type / creator:** Second-draft community policy from KDE Plasma Workspace. **Link:** KDE work item. The work item does not identify an individual proposer. [6]
- **Recommended by:** Tobi, who endorsed the linked discussion with “This is the way” and argued that code should be accepted on merit while a person remains accountable, regardless of whether it was typed or generated.

Link: Tobi’s endorsement. [7]

- **Key takeaway:** The draft’s “golden rule” is “Don’t be lazy.” It requires a human in the loop, rejects throwaway or “vibe-coded” changes the contributor does not understand, generally discourages LLM-generated prose, and requires accuracy checks when LLMs are used for debugging or research. [6]
- **Why it matters:** Unlike a general exhortation to “use AI responsibly,” this gives reviewers concrete criteria: human judgment, understanding, verification, and sustainable work. It is explicitly a proposal, not a final policy.

Two product-leader book picks

The Lenny interview’s guest, Peter Sellis, was asked for the books he recommends most. He named *How to Get Filthy Rich in Rising Asia*, *When Genius Failed*, and *Einstein’s Dreams*; the latter two came with the most useful context.

Link: source interview. [8]

- **** *When Genius Failed* — book; creator not identified in the interview.**** Sellis describes it as the rise and fall of Long-Term Capital Management in late 1998 and a nearly averted crash. That makes it the strongest of the three as a decision-making and risk resource, rather than a generic business recommendation. [8]
- **** *Einstein’s Dreams* — book by Alan Lightman.**** Sellis says he gives it to many consumer product managers. The useful signal is the audience: he treats a literary/scientific work about time as relevant to product people, not merely as a personal favorite. [8]

A lighter personal pick

Professor T

- **Content type / creator:** TV series; the creator is not identified in Paul Graham’s post. **Link:** Graham’s recommendation.
- **Recommended by:** Paul Graham, who says he and Jessica have been watching it and calls it “really good.”
- **Key takeaway:** Graham frames it as a possible substitute for readers who wish Arthur Conan Doyle had written more Sherlock Holmes stories. [9]
- **Why it matters:** This is a taste recommendation rather than a technical resource, but it is unusually specific and personal—the kind of organic signal worth preserving separately from the research-heavy picks.

Sources

1. X post by @martin_casado

2. HYDRA: the kernel of a multiprocessor operating system
3. Labels and event processes in the asbestos operating system
4. X post by @chamath
5. X post by @ylecun
6. Proposed KDE LLM guidelines (#187) · Issues · Plasma / Plasma Workspace · GitLab
7. X post by @tobi
8. 90 minutes of unfiltered product advice from Snap and Discord's product chief | Peter Sellis
9. X post by @paulg