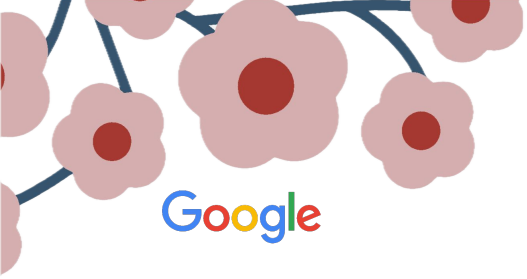


Google

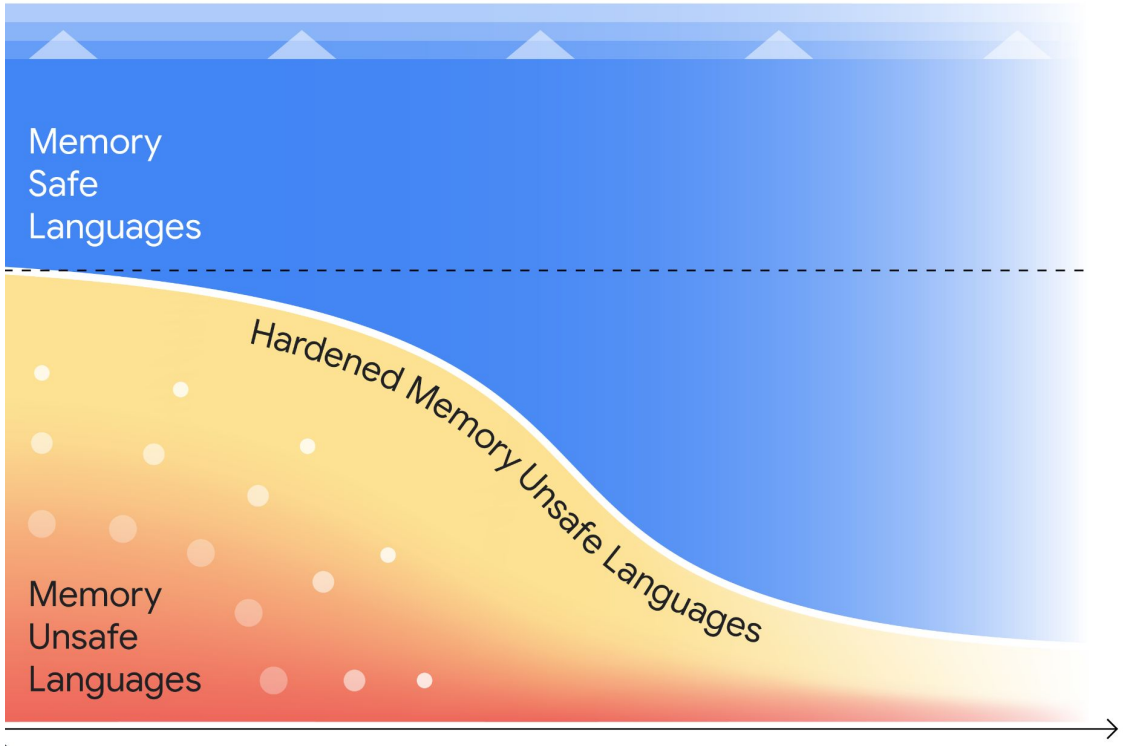
Memory Safety at Scale: An Industry Perspective

Alex Rebert
Memory Safety @ Google
CHERI Blossoms 2026

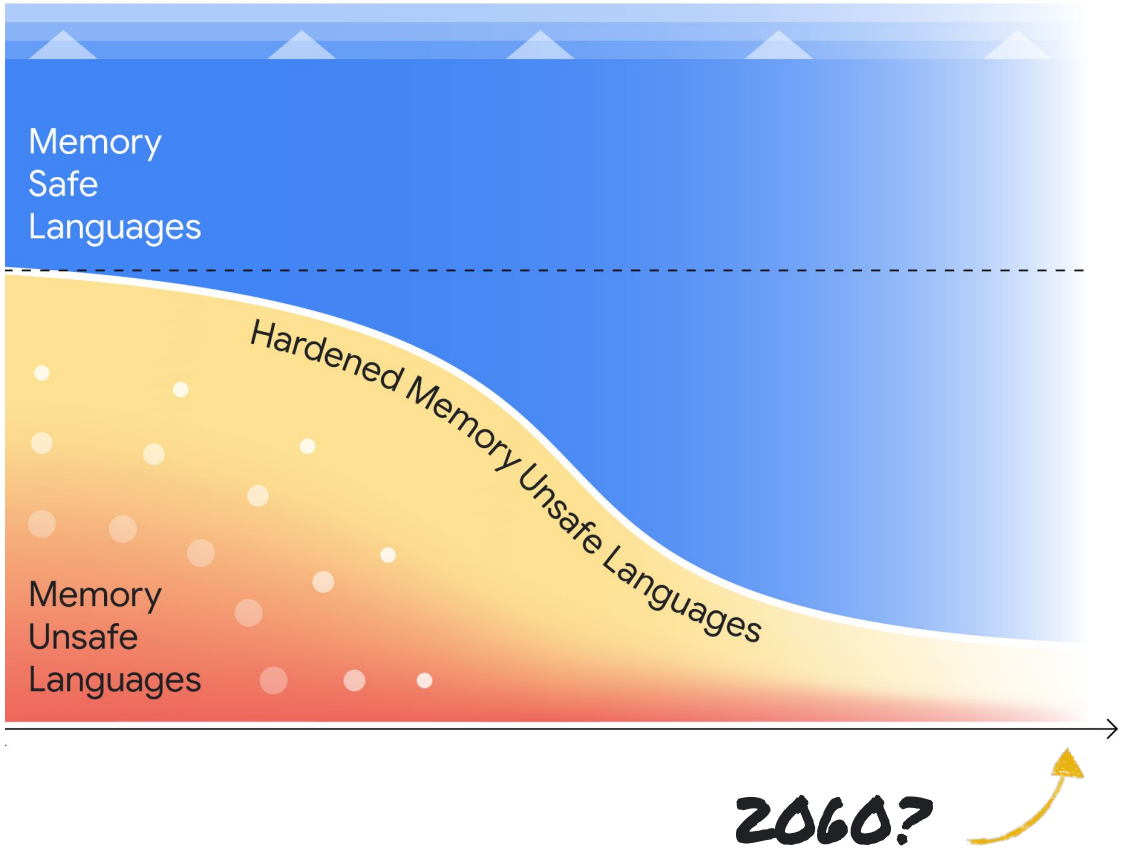
“Unsafe at Any CPU Speed:

The Designed-in Dangers of Technology and What We Can Do About It”

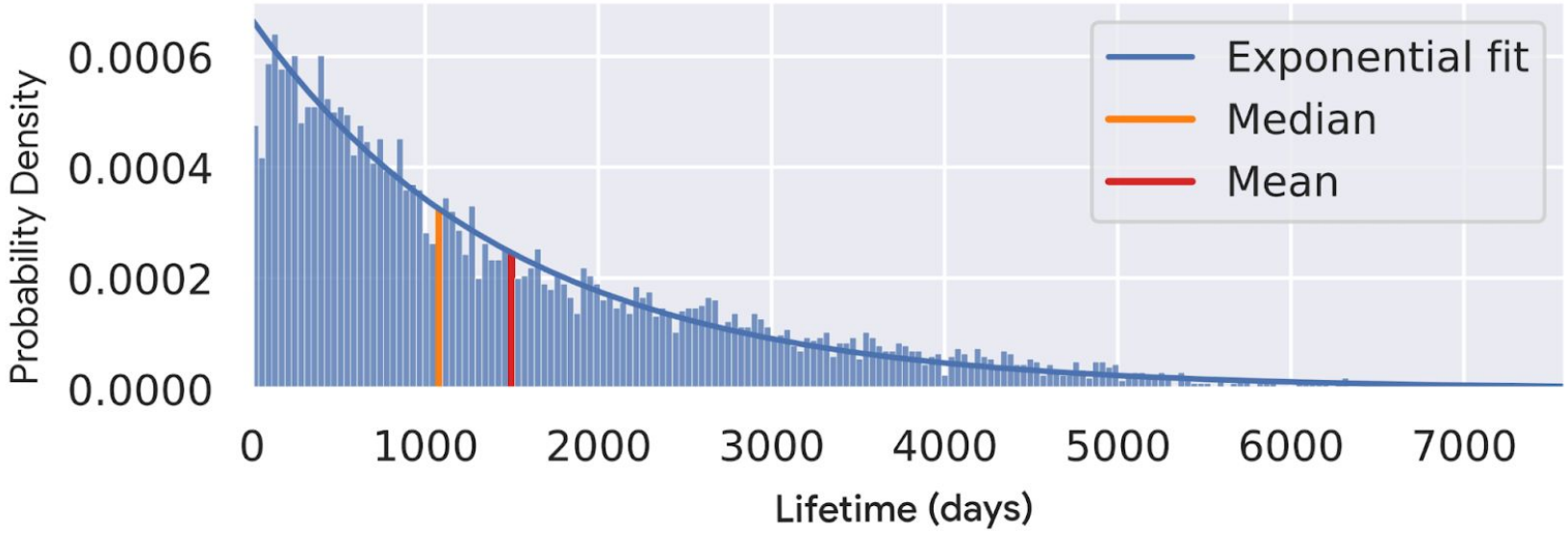
Secure by Design: The Software-First Shift



How long will this take?

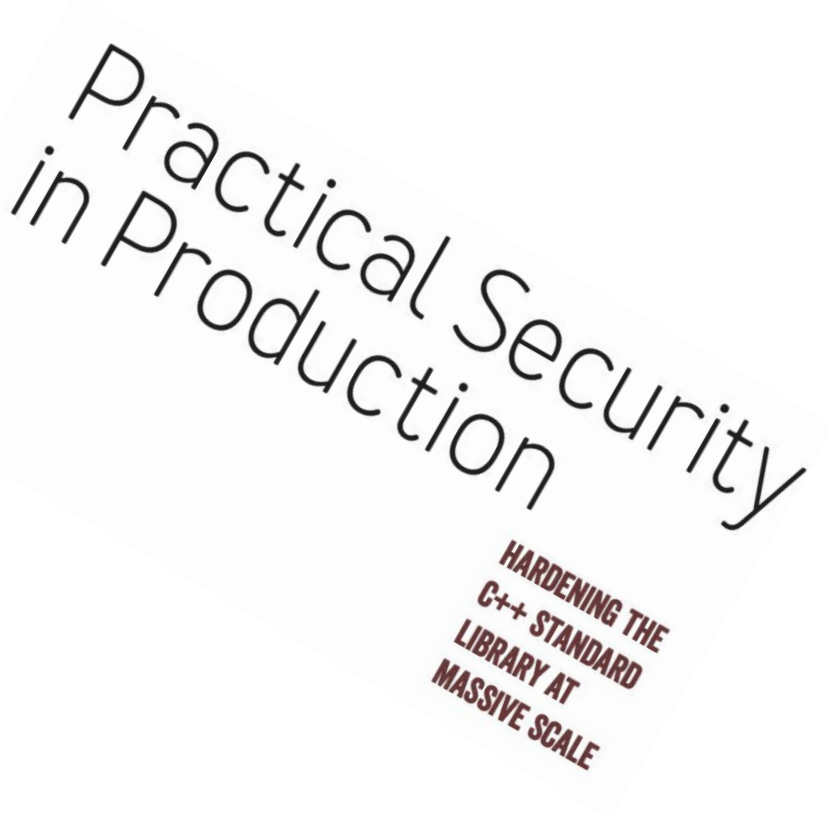


Force 1: Vulnerabilities have a Half-Life

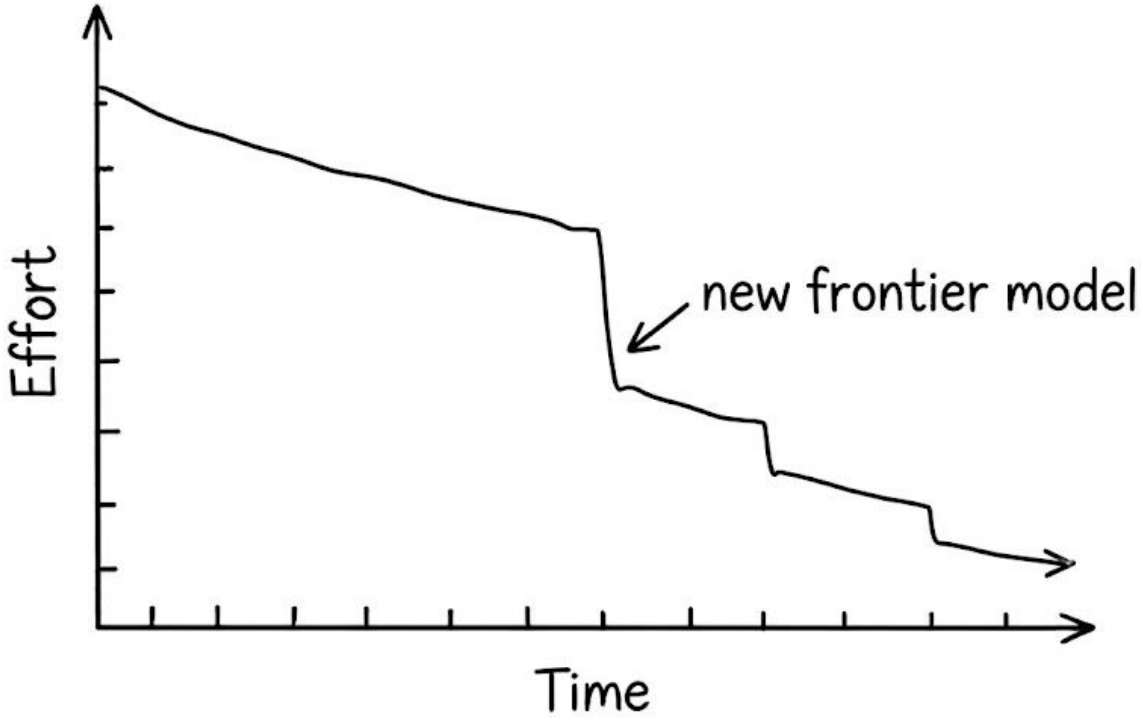


From "How Long Do Vulnerabilities Live in the Code? A Large-Scale Empirical Measurement Study on FOSS Vulnerability Lifetimes". Alexopoulos et al. USENIX Security 22.

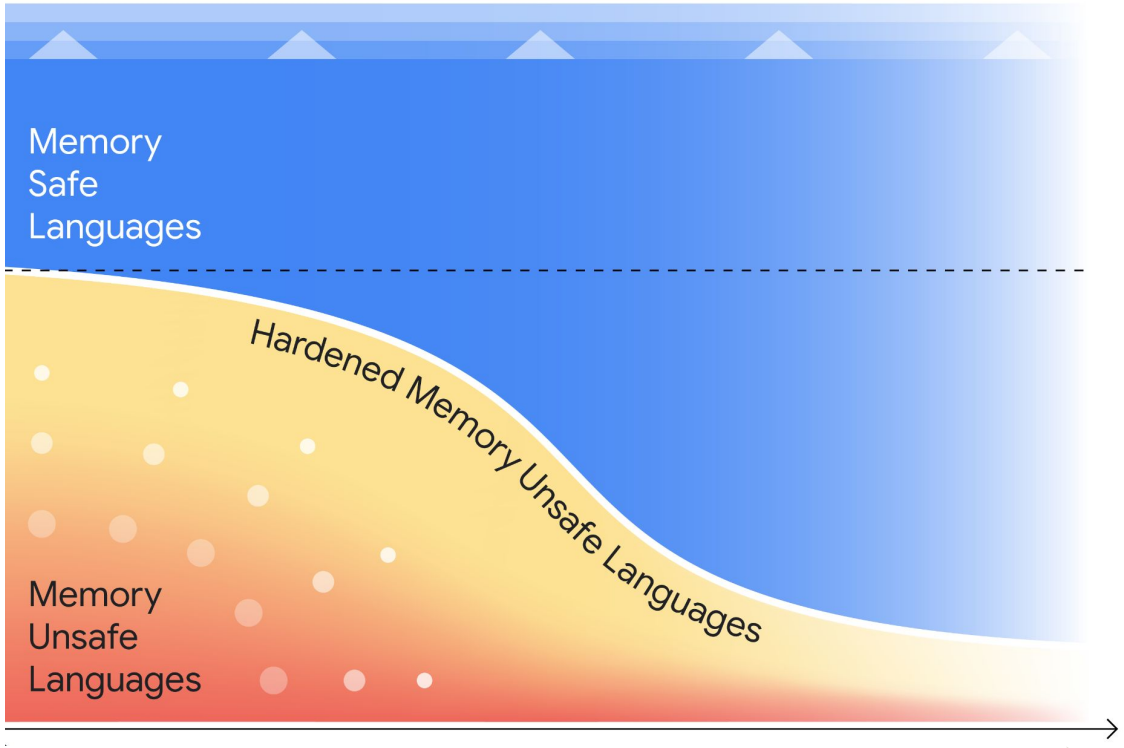
Force 2: Affordable Spatial Hardening



Force 3: AI translations



10-100x Software Acceleration?



Remaining gap: Temporal Safety

FIL-C PERFORMANCE

*"between **1x and 4x** as many cycles..."*

— djb

FIL-C MEMORY OVERHEAD

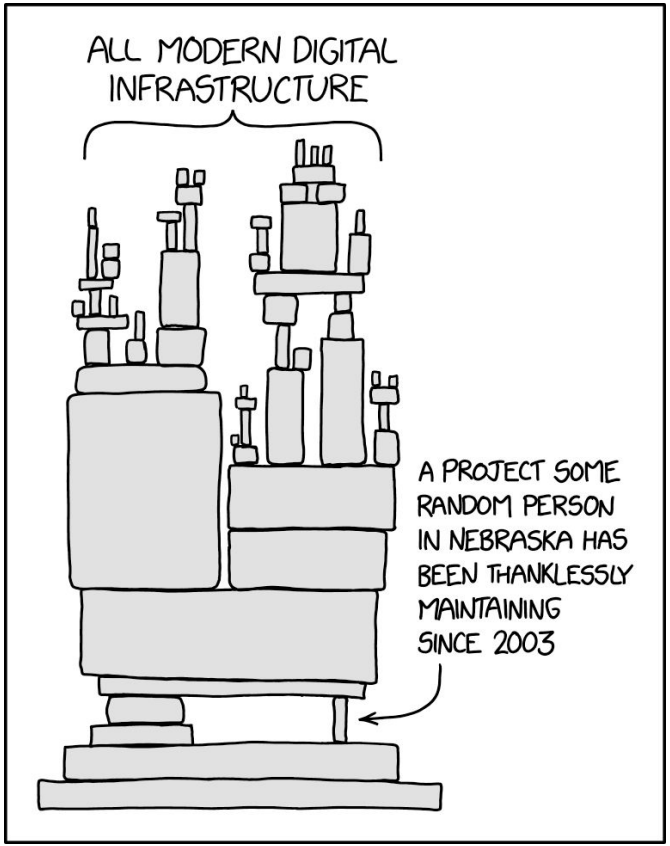
*"looks to be more like **3-6x**"*

— Graydon Hoare

Rust-style safety model for C++ 'rejected' as profiles take priority

Safe C++ proposal author claims that 'will not ever work'

Remaining gap: Supply Chain



https://xkcd.com/2347/


CHERI

World 1: AI does not totally change the memory safety game

The Premise

AI progress levels off. Mass-scale rewrites of the open-source ecosystem remain out of reach. We can harden 1st-party code, but rely on a massive C/C++ supply chain, particularly for large/critical TCBs like OSes and language runtimes

CHERI's Value

Provides **spatial safety** for legacy code we don't own and cannot change.

The Challenge: Temporal Safety

Hardware temporal safety (e.g., memory scanning) *prototypes* carry a heavy tax. We may need architectural breakthroughs for mass enablement including in constrained or high-performance environments.

World 2: The Great Rewrite

01

JITs & Language Runtime

Rewriting JITs like V8 in memory-safe languages will not solve all memory safety problems.

02

OSes & Firmwares

Low-level code have to interact with a bunch of hardware components, where language-level safety properties cannot be enforced.

03

Cross-Language FFI / Compartmentalization

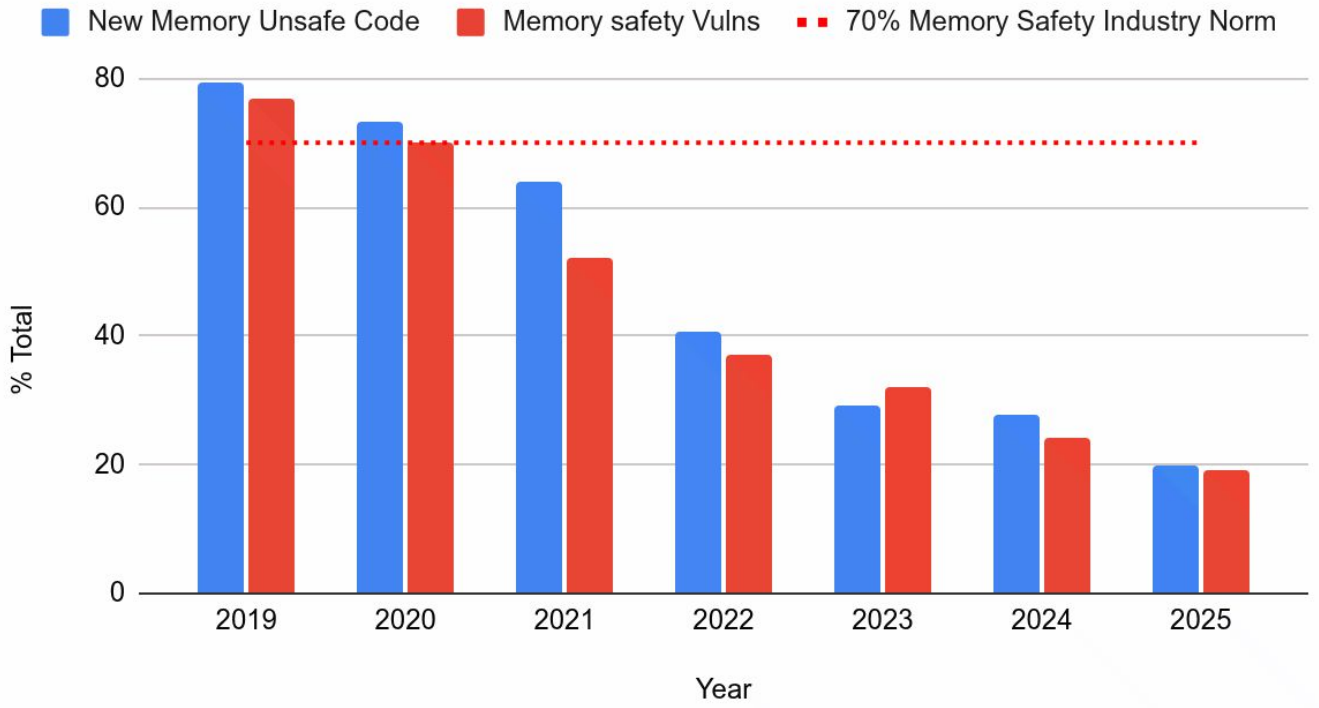
Realistically, we will continue to rely on C/C++ even in after "the big rewrite" (parts of kernel, OSS libraries, ...).

Thank you

Alex Rebert

Memory Safety @ Google

Plot Twist 1: Vulnerabilities have a Half-Life



https://security.googleblog.com/2025/11/rust-in-android-move-fast-fix-things.html