Building for Safety in a Security and Feature Focused World

Chuck Wolber
Associate Technical Fellow
The Boeing Company

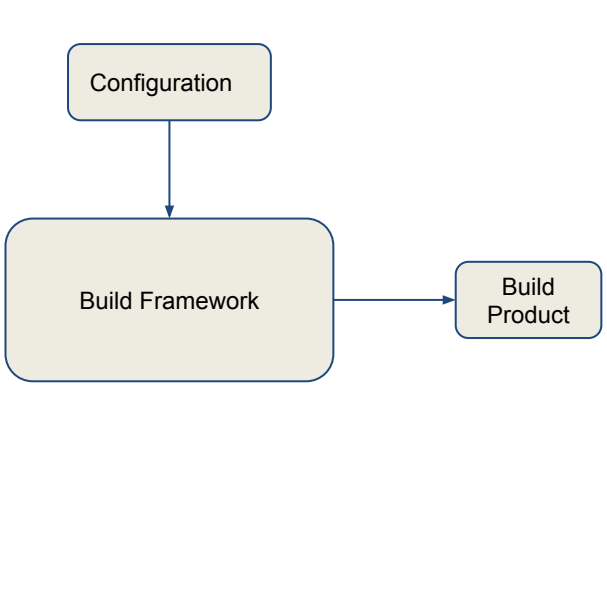
whoami(1)

- B.S. Mathematics, Washington State University
- B.S. Computer Science, Seattle University
- Contributor to several early editions of Red Hat Linux Bible
- Co-author of Linux Toys (Wiley Publishing)
- President and BDFL - Tacoma Linux Users Group
- 20+ Years – Boeing Software Engineer
  - Inducted to Technical Fellowship in 2022.
  - Developed a (DO-178B/C) design certified Linux platform delivered on most Boeing Aircraft

Punchline First

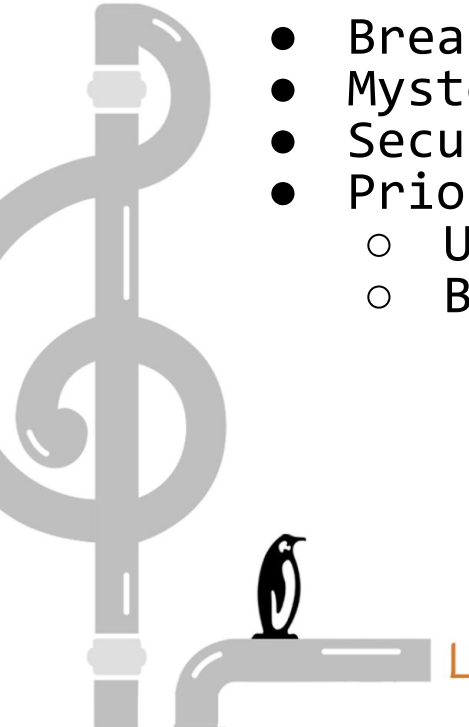
Security^H^H^H^H^H^H^H^H**Safety is a diverse form scrutiny that will broadly improve OSS.**

Build System



Loose Federation Problems

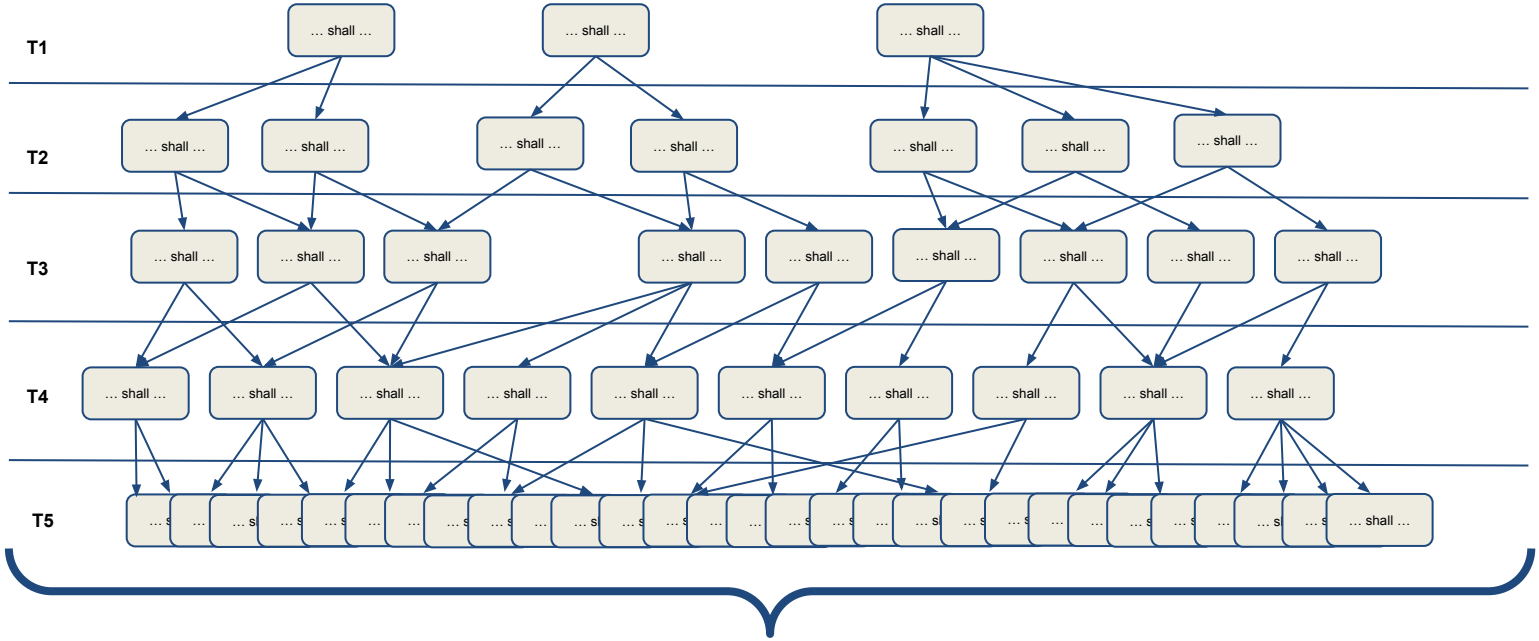
- Breaking change upstream…
- Mysterious incompatibilities…
- Security bug in unmaintained branch…
- Priority mismatch…
  - Upstream ❤️ features!
  - Build ❤️ stability!



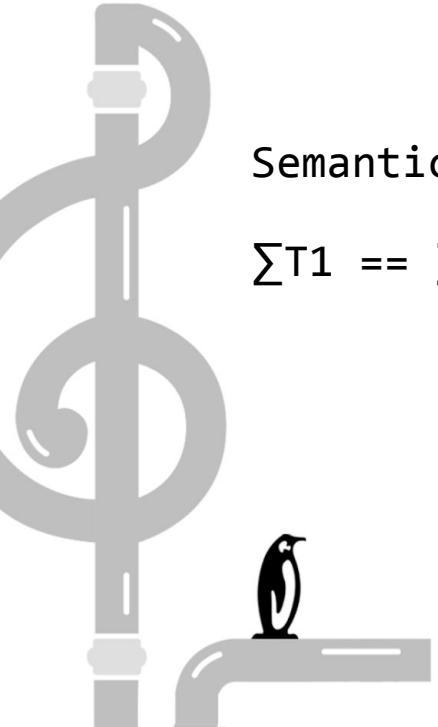
Safety

Safety	=	Specified Function + Deterministic Behavior
Safety Engineering	=	Hazard Assessment + Design Mitigation + Implementation Assurance

Design Expression



Implementation Assurance



Semantically speaking…

∑T1 == ∑T2 == ∑T3 == ∑T4 == ∑T5 == Implementation

Safety Problems

- Loose Federation Problems

Plus…

- Design expression is resource intensive.
- Implementation assurance is resource intensive.
- Assessing change impact is difficult.

Now What?

"The kernel gets nine patches per hour, requirements are never going to be a priority." - GKH (Paraphrased)

- Design Expression
  - SPDX FuSa - Machine readable safety model, including requirements!
- Implementation Assurance
  - llvm-cov - Patches submitted to Linux Kernel
- Assessing Change
  - Delta Kernel - github.com/elisa-tech/delta-kernel