Aerospace Working Group

2026 Annual Update

Matthew Weber & Dr. Martin Halle

FIX to point at new slides

ELISA
Enabling Linux in Safety Applications

Aerospace · Automotive · Linux Features
OS Engineering Process · Safety Architecture · Systems · Tools
Lighthouse · Space Grade Linux

Matthew Weber

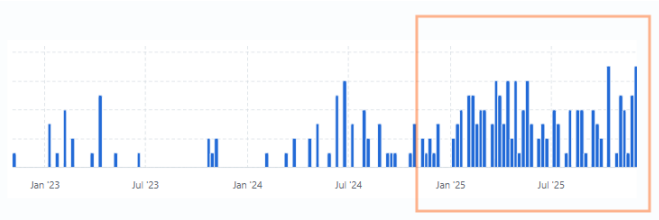
Associate Technical Fellow @
The Boeing Company

Aerospace Working Group

Charter:

"… shall develop use cases to inform and influence Linux architecture and related tools, work to derive technical requirements for avionics operating systems, and seek to enhance and expand avionics software lifecycle processes, practices, and tools to enable use of Linux in avionics systems that are certified to high design assurance levels."

2025 Contributions



- Active participation: (2024 vs. 2025)
  - 46 → 48 unique members attended meetings
    - 23 → 26 meetings (Combined Aerospace + SGL)
    - Aerospace average meeting attendance is 8.91 → 8.7
  - Added a weekly use case demo development call – Average attendance of 4.5
  - Overall 295 commits from 9 contributors

- Thank you for your contributions of time and knowledge this year!

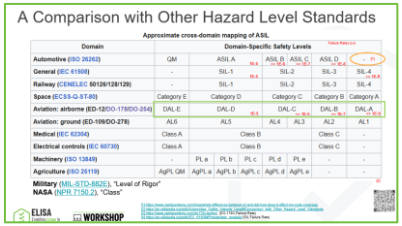
  Aerospace Corporation, Astro Mechanica, Barcelona Supercomputing Center, Boeing, Collins Aerospace, Coros Space, Czech Aerospace Research Centre, Devstringx Technologies, DLR (German Aerospace Center), ESA (European Space Agency), ETAS GmbH, Hamburg University of Technology, KBR @ NASA Ames Research Center, L3Harris, Timesys/Lynx Software, Microchip Technology Inc., NASA (various centers including Ames, Goddard, Langley), NXP Semiconductors, OpenEmbedded, Rapita Systems, Red Hat, Sony, Space Cubics, Stoke Space, TII (Technology Innovation Institute), Voyager, Vorago Technologies, Wind River

- Special thanks to those contributing to paper and demo development!

  Barcelona Supercomputing Center, Boeing, Collins Aerospace, Coros Space, Devstringx Technologies, DLR (German Aerospace Center), Hamburg University of Technology, KBR @ NASA Ames Research Center, Timesys/Lynx Software, Wind River

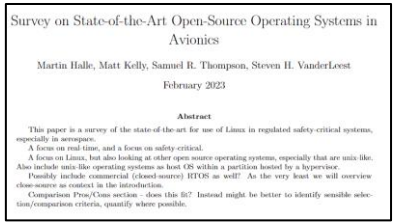
2025 Achievements

- Presented at the Munich workshop
  - “Industry Safety Levels vs Use Cases”



  - “Research questions and publication directions of Aerospace WG”

- Restarted a monthly “Industry Paper” call


Survey on State-of-the-Art Open-Source Operating Systems in Avionics

Martin Halle, Matt Kelly, Samuel R. Thompson, Steven H. VanderLeest

February 2023

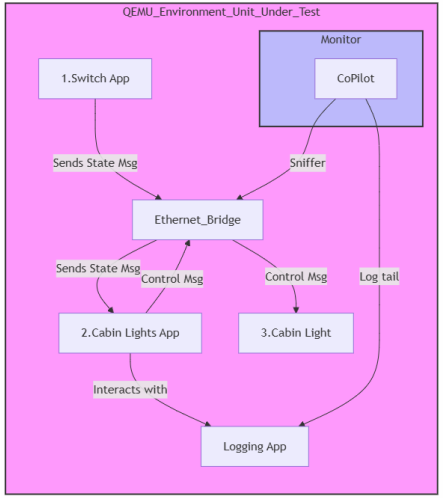
Abstract

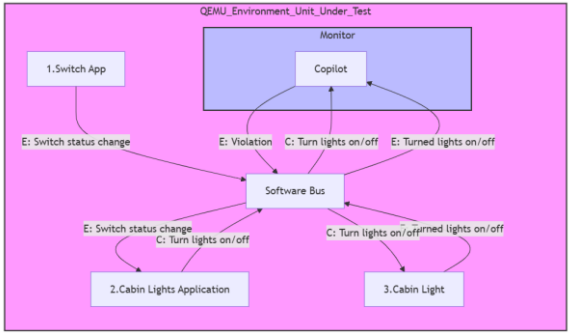
This paper is a survey of the state-of-the-art for use of Linux in regulated safety-critical systems, especially in aerospace.
A focus on real-time, and a focus on safety-critical.
A focus on Linux, but also looking at other open source operating systems, especially that are unix like. Also include unix-like operating systems as host OS within a partition hosted by a hypervisor.
Possibly include commercial (closed-source) RTOS as well? As the very least we will overview close-source as context in the introduction.
Comparison Pros/Cons section - does this fit? Instead might be better to identify sensible selection/comparison criteria, quantify where possible.

- Iain Galloway(NXP) presented on ASIL-D activities with robotics
- ELISA SystemWG presented on reference systems work and SoDev

2025 Achievements

- Completed Cabin Lights demo!
  - Requirements
    - The Cabin Lights system shall turn lights on in less than 500 ms of the light switch turning on.
    - The Cabin Lights system shall turn lights off in less than 500 ms of the light switch turning off.
  - Design
    - Switch, Server and Actuator(Light)
    - Ethernet and system logging are used
    - QEMU ARM64 and minimal Linux Kernel + Busybox
  - Test / Demo approach
    - Applications are paired with NASA CoPilot monitoring of logs / package





- Extended the Cabin Lights demo with NASA Core Flight System (cFS)

2025 Achievements

- Evaluated how to capture distinguishing system requirements, i.e.," What makes a product aerospace vs. just embedded?”
  - Discussed Carrier Grade Linux as an example
  - Created a template for capturing “aspects” of a product’s profile
  - Organizing by safety standard and level
  - Capturing examples at each level

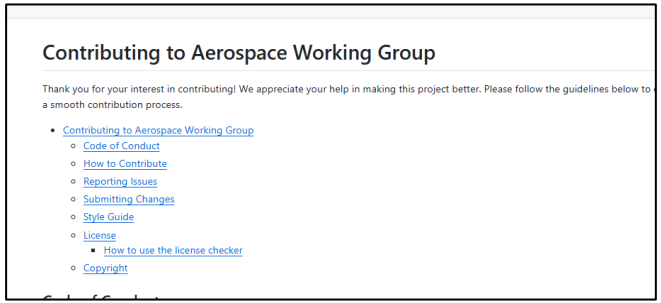
Template - "name of notional system"

Below is an example table of product "aspects" and examples of what "details" might be considered when defining a new "product profile."

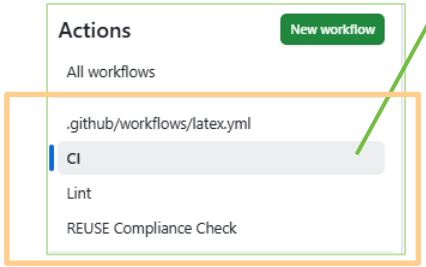
Product Aspect(s)	Details	Notes
Operational duration:		
Userdata update cycle:	rate / scale	
Platform update cycle:	rate / scale	
Software Level:	A/B/C/D/E	
Security:	e.g., Security Assurance Level (SAL)- DO-326/365a, layered, hw based, lifecycle rigor X	Clarification of constraints
App Capability:	e.g., POSIX Apps, ARINC Apps, Web Apps, Applets, Bare metal, Scripting (Interpeter)	
System Constraints:	e.g., deterministic, simple, radiation hardened, reliability	
Architecture Constraints:	e.g., Redundancy, dissimilarity, monitoring, real-time	Could drive additional HW / specific configurations within a system. Real-time needs details on scheduler and expected behavior of events(low/med/hard)
Protocols:	Networking RFCs, ARINC, Streaming video encodings,	
IO:	e.g., ARINC (429/717/664/B25...), MIL (1553...), IEEE/SAE (TSN 802.1DP...), serial, CAN bus, Ethernet, PCIe, USB, Spacewire	
Dataload/fieldload:	e.g., ARINC-based mediaset via ARINC615a dataloader, ARINC615-3, Commercial OTA, Floppy disk	
OS standards:	e.g., ARINC653 modular partitioned operating system, Tailored / customized "carrier grade", Firmware vs. OS, SELinux, POSIX, Engine-like environments (eBPF, WASM, Container engine), Orchestration(kubernetes)	
OS footprint:	Small	below 200k SLOC
OS Scheduling:	E.g. scheduler used (real-time?), ARINC653 Apps/tasks, processes, threads, MMU-isolated, namespaced	
Boot-up time:	within seconds / milli-seconds	Frame this as (warm vs. cold start)
Fault handling:	e.g., detection, remediation, tree showing domains/isolation/cascading, BITE, logging, reporting upstream, FMEA	Development analysis / design / operational
Languages:	C, ADA	(Future) RUST
Storage:	512MB - low GB	
Memory:	512MB - low GB	
Memory allocation:	static / dynamic	Impacted by system state? When allowed to allocation, can you free, etc?
Worst Case Execution Time (WCET):	Knowledge about WCET for OS / user-code functions, known max. induced interferrence (e.g., on multi-core)	
CPU Performance:	MHz/FLOPS/DMIPS/Optimization level	
CPU Cores:		
Tool Qualification:	NA/Possible	(To what level / background details)

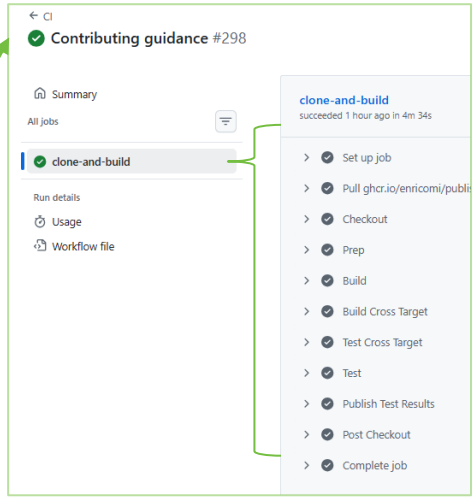
2025 Achievements

- Established our Contributions guidance and Licensing



- Matured CI workflows on push
  - Stages to Build, Test, and Report (Block merge)
  - Instrumented emulated testing
  - Sandboxed tools for reproducibility
  - Linting of review material
  - License check

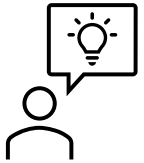




2026 Activities

- Expand on capturing products with profile template
  - Create NASA profiles → Aids Space Grade Linux
  - Tie product profiles to use cases → Drive future reference system
  - Elaborate on fault tolerance and availability aspects
- A reference system (QEMU & HW) for benchmarking (a subset of current demos)
- Find common topics with Space Grade Linux
  - E.g. reference system definition, footprint, reliability
  - Industry Survey paper
- Submit abstract to a conference for at least one Industry paper

How to Get Involved



Join our monthly call(s)

- 2nd Thursday – “General Topics”
- 3rd Thursday – Space Grade Linux SIG
- 4th Thursday – “Industry Papers” session
- Weekly (Friday) – “Use Case” testing call

Register for calendar invite

https://elisa.tech/community/meetings/

Mailing List - https://lists.elisa.tech/g/aerospace

Discord - https://chat.elisa.tech/

GitHub - https://github.com/elisa-tech/wg-aerospace

- View meeting minutes
- Use case and whitepaper documents
- Use case Demos