Aspects of Dependable Linux Systems

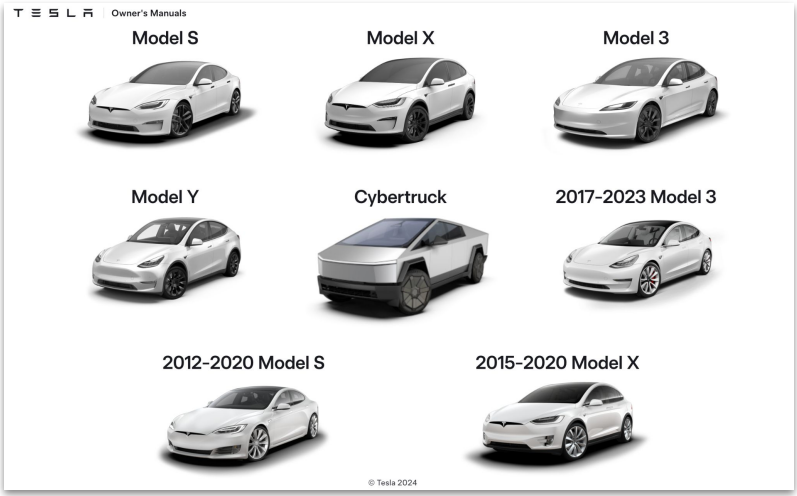
Kate Stewart, Linux Foundation
Philipp Ahmann, Etas GmbH (BOSCH)

Safe Systems with Linux MC

LINUX PLUMBERS CONFERENCE | Vienna, Austria
Sept. 18-20, 2024

ELISA
Enabling Linux in Safety Applications

Linux is being used in Safety Critical Systems today...



source: https://www.tesla.com/ownersmanual

source: https://www.spacex.com/mission/

What is functional safety?

Definition of Safety

The freedom from unacceptable risk of physical injury or of damage to the health of people, either directly, or indirectly because of damage to property or the environment.

Definition of Functional Safety

The part of safety that depends on a system or equipment operating correctly in response to its inputs.

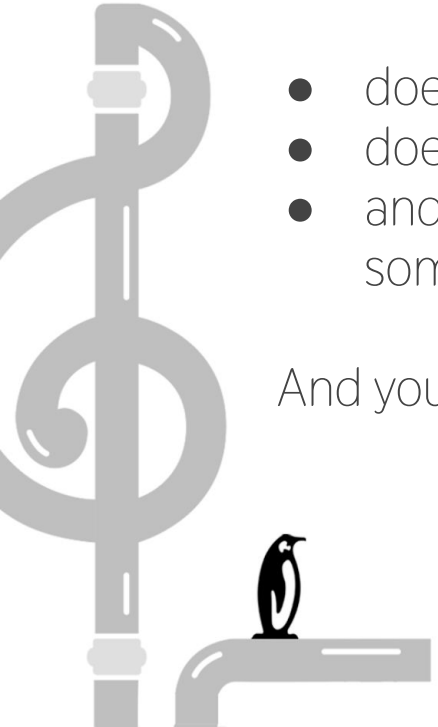
Detecting potentially dangerous conditions, resulting either in the activation of a protective or corrective device or mechanism to prevent hazardous events or in providing mitigation measures to reduce the consequences of the hazardous event.

In Functional Safety you expect...

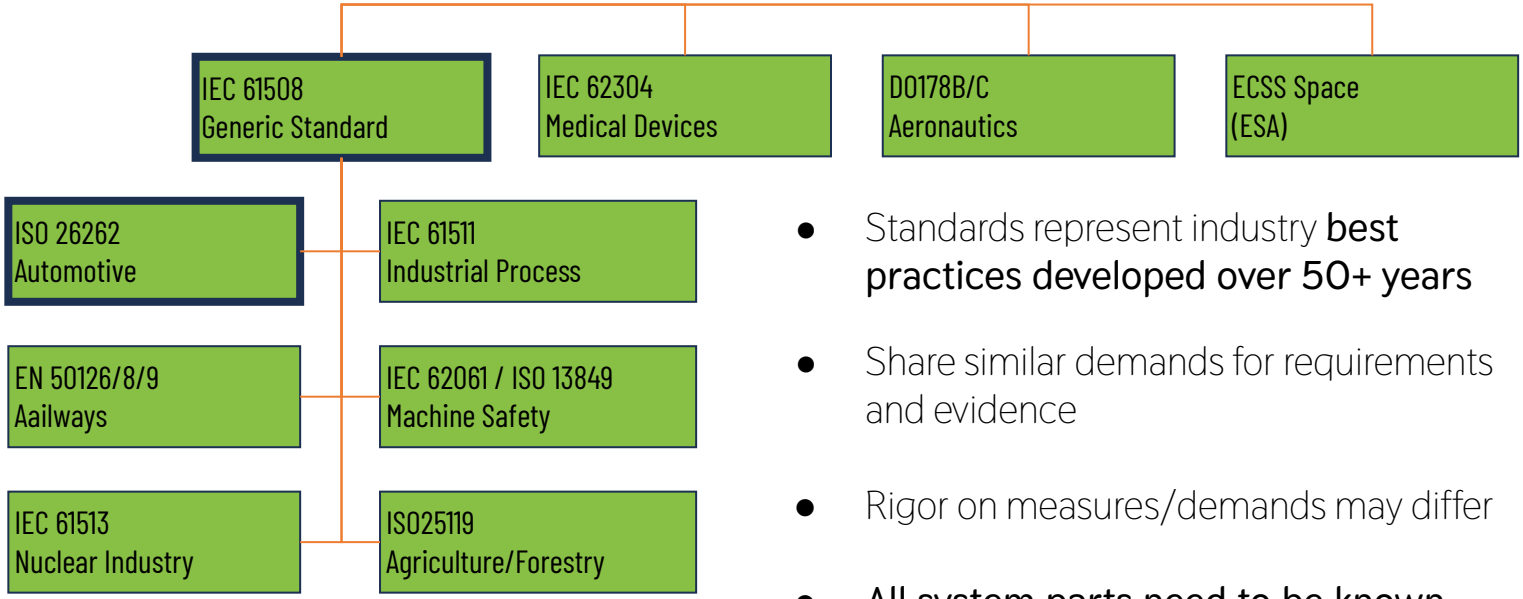
...that the software:

- does behave as specified,
- does not interfere or impair other system components
- and all possible erroneous events are addressed somehow or somewhere.

And you have sufficient evidence to prove this.



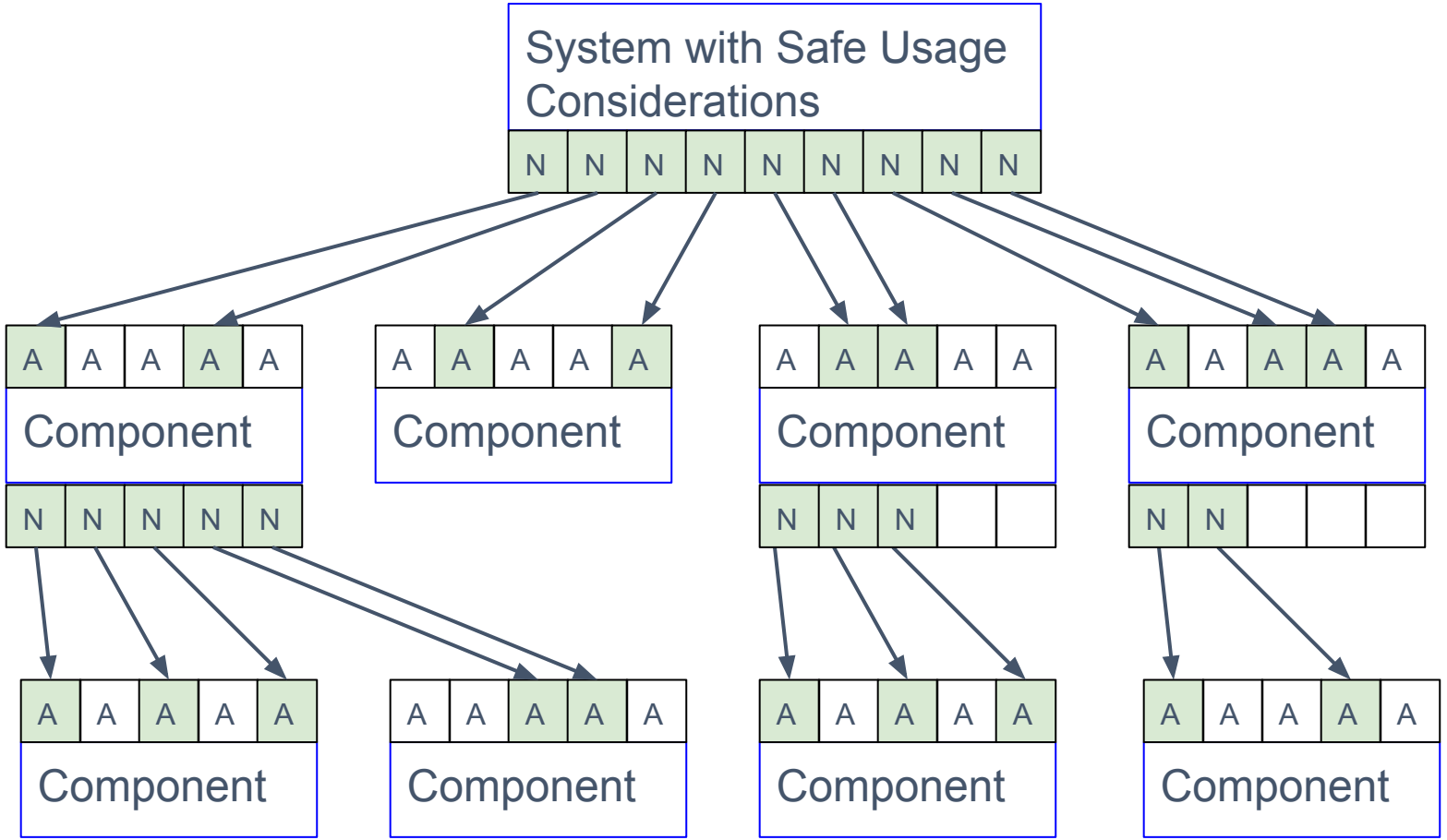
Samples of safety (integrity) standards

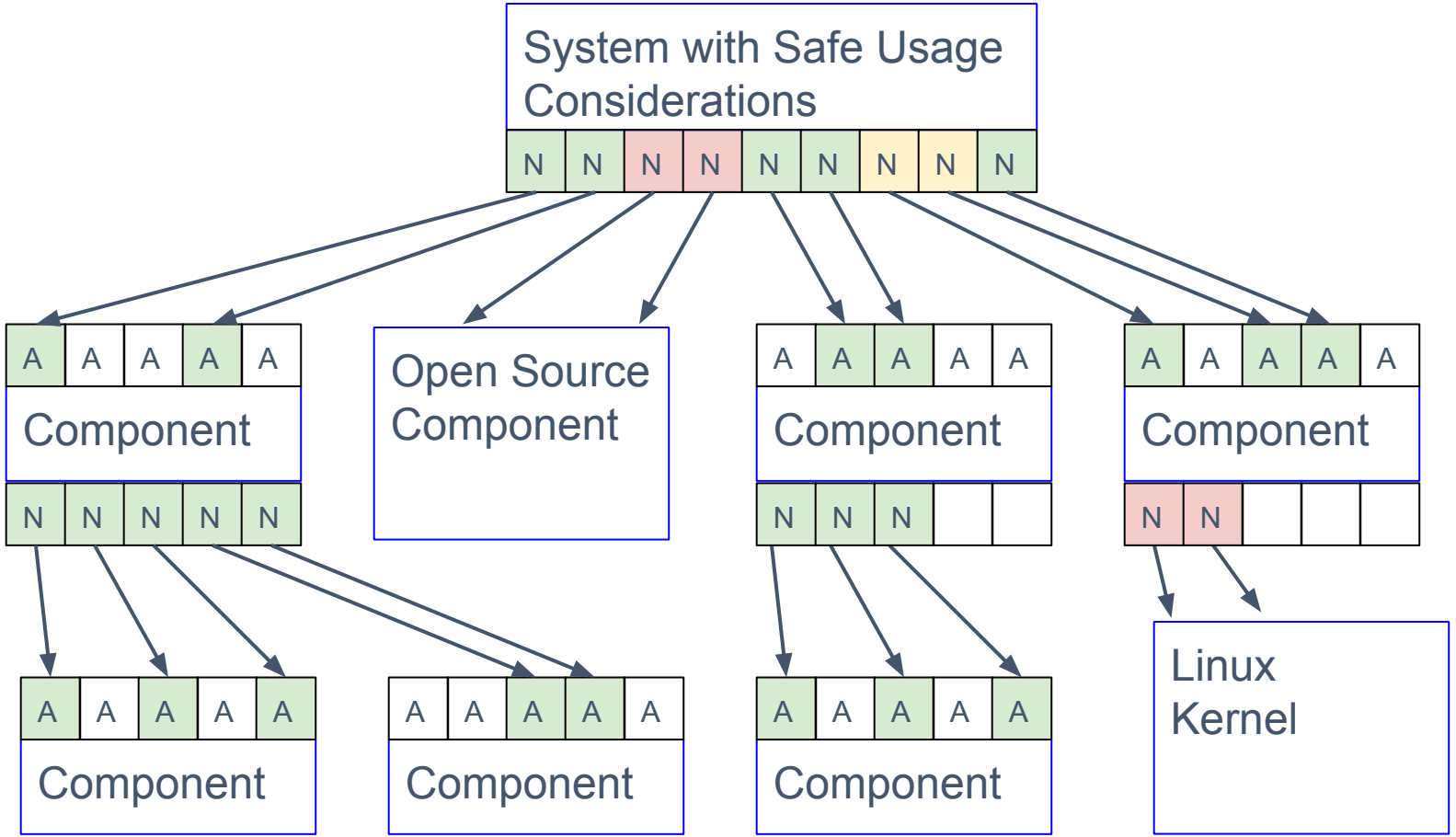


- Standards represent industry **best practices developed over 50+ years**
- Share similar demands for requirements and evidence
- Rigor on measures/demands may differ
- **All system parts need to be known, tested and managed**

Standards seek to increase system quality

- Requirements (being explicit about assertions)
- Testing & Evidence
- Documentation
- Traceability


System with Safe Usage Considerations
N N N N N N N N N
A A A A A
Component
A A A A A
Component
A A A A A
Component
A A A A A
Component
N N N N N
N N N
N N
A A A A A
Component
A A A A A
Component
A A A A A
Component
A A A A A
Component


System with Safe Usage Considerations
N N N N N N N N N
A A A A A
Component
N N N N N
Open Source Component
A A A A A
Component
N N N
A A A A A
Component
N N
Linux Kernel
A A A A A
Component
A A A A A
Component
A A A A A
Component

Challenge with Safe Usage of Linux Kernel

Each patch has a reason for being added to tree - "what" & "why"

- frequently contained in patch series overview, but may be part of email discussion.
- Understanding "what" the code should do, is considered as a "**requirement**" on a component (like the kernel) when doing functional safety system analysis.
- Testing the functionality for when it **works**, and when it **does not work** is needed as "evidence" that is required to assess "Safe Usage".

Challenge: Linux Kernel does not have a way of systematically capturing the "what" code is expected to do in a machine readable form.

If the "assertions about the code" (may be referred to as specifications or requirements) are reverse engineered by others, where should they be stored, so they can be reviewed by maintainers and other experts?

What mechanisms should be used to link the code & tests to these requirements?

An afternoon towards "Safe Systems with Linux“

Addressing:

- Systems
- Static Analysis
- Code Coverage
- Aequirements / Traceability

- SBOM
- System Engineering
- Documentation

15:00	**Aspects of Dependable Linux Systems** "Hall N2", Austria Center	*Kate Stewart et al.* 15:00 - 15:15
	Verifying the Conformance of a VirtIO Driver to the VirtIO Specification "Hall N2", Austria Center	*Matias Vara Larsen* 15:15 - 15:45
	ks-nav "Hall N2", Austria Center	*Alessandro Carminati* 15:45 - 16:00
16:00	**Source-based code coverage of Linux kernel** "Hall N2", Austria Center	*Wentao Zhang et al.* 16:00 - 16:15
	BASIL development roadmap "Hall N2", Austria Center	*Luigi Pellecchia* 16:15 - 16:30
	Break "Hall N2", Austria Center	16:30 - 17:00
17:00	**Enabling tooling independent exchange of Requirements and other SW Engineering related information with the upcoming SPDX Safety Profile** *Nicole Pappler*	
	Throwing Cinderblocks at Safety Engineering "Hall N2", Austria Center	*Chuck Wolber* 17:25 - 17:50
18:00	**Improving kernel design documentation and involving experts** "Hall N2", Austria Center	*Gabriele Paoloni* 17:50 - 18:10
	Discussion of Next Steps "Hall N2", Austria Center	*Kate Stewart et al.* 18:10 - 18:30

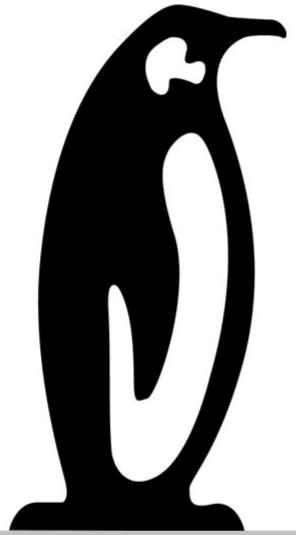
LINUX PLUMBERS CONFERENCE | Vienna, Austria
Sept. 18-20, 2024

ELISA
Enabling Linux in Safety Applications

Safety Critical Systems

“Assessing whether a system is safe, requires understanding the system sufficiently.”

- Understand your system element within that system context and how it is used in that system.
- Select system components and features that can be evaluated for safety.
- Identify gaps that exist where more work is needed to evaluate safety sufficiently.



Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024