The ELISA Project

Enabling Linux in Safety Applications

Philipp Ahmann, Robert Bosch GmbH

Embedded IoT
Linux@Bosch

CIVIL INFRASTRUCTURE PLATFORM

OPEN CONTAINER INITIATIVE

APERTIS

AUTOMOTIVE GRADE LINUX

eclipse SDV

kanto

SPDX

KUKSA

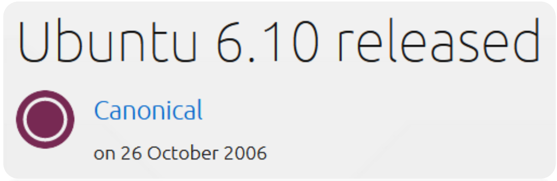
SOAFEE

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

OSS Review Toolkit

whoami

- **Technical business development manager for embedded open source - Robert Bosch GmbH**
- **Technical Steering Committee Chair & WG Lead - Linux Foundation's ELISA project**
- **15 years+ Linux user (and open source enthusiast)**
- **10 years+ Linux in Automotive (Infotainment)**



https://oss.bosch-cm.com/

Photo by Ashutosh Dave on Unsplash

Linux in Safety Critical Systems

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

Understand Linux within that system context and how Linux is used in that system.

Selecting Linux components and features that can be evaluated for safety.

Identifying gaps that exist where more work is needed to evaluate safety sufficiently.

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Challenge: Linux in Safety-Critical Systems

The Linux kernel has:

- Large Development Ecosystem
- Security Capabilities
- Multi-Core Support
- Unmatched Hardware Support
- Many Linux Experts at all levels available

Traditional safety-critical OS has:

- Hard Real-time Capabilities
- Proven Safety-compliant Development Process
- …

Can these differences be tackled?

ENABLING LINUX IN SAFETY APPLICATIONS

Understanding the Limits

The collaboration…

- *cannot engineer* your system to be safe.
- *cannot ensure* that you know how to apply the described process and methods.
- *cannot create* an out-of-tree Linux kernel for safety-critical applications. (Remember the continuous process improvement argument!)
- *cannot relieve* you from your responsibilities, legal obligations and liabilities.

But…

ELISA provides a path forward and peers to collaborate with!

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Premier Members

BMW GROUP
BMW Car IT

BOEING

intel

Red Hat

TOYOTA

General Members

ADIT

AICC 国汽智控

AISIN

arm

斑马智行
Powered by AliOS

BOSCH

Codethink

Elektrobit

地 平 线
Horizon Robotics

HUAWEI

LINUTRONIX
LINUX FOR INDUSTRY

LOTUS NYO

SAIC 上汽集团
SAIC MOTOR

SUSE

SUZUKI

WNDRVR

XPENG

ZTE

Associate Members

AUTOMOTIVE GRADE LINUX

OTH
OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Industry Support

CIVIL INFRASTRUCTURE PLATFORM

OSADL
Open Source Automation Development Lab eG

UL

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

“The mission of the project is to define and maintain a common set of elements, processes and tools that can be incorporated into Linux-based, safety-critical systems amenable to safety certification.”

“The scope of the project includes software and documentation development under an OSI-approved license supporting the mission, including documentation, testing, integration and the creation of other artifacts that aid the development, deployment, operation or adoption of the project.”

from the technical charter

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

ELISA Working Groups - Deliverable grouping (based on mission)

- Elements / Software
- Processes
- Tools
- Documentation

ENABLING LINUX IN SAFETY APPLICATIONS

Working groups (WGs)

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Horizontal Working Groups

Safety Architecture

Red Hat

Open Source Engineering Process

Codethink

Linux Features

mobileye / intel

Systems

BOSCH

Tool investigation & Code Improvement

Elektrobit Codethink

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Vertical Use Cases

New!

Aerospace

BOEING

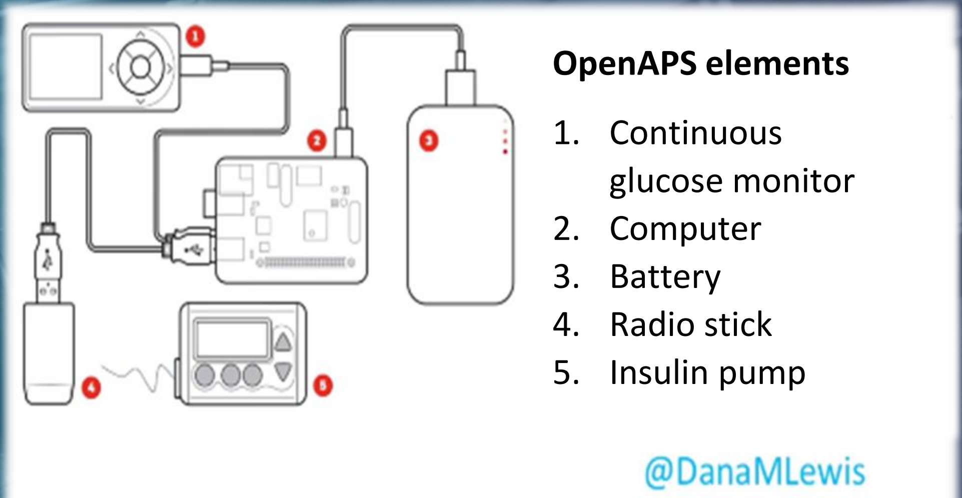
Automotive

BOSCH

Medical Devices

LINUX FOUNDATION

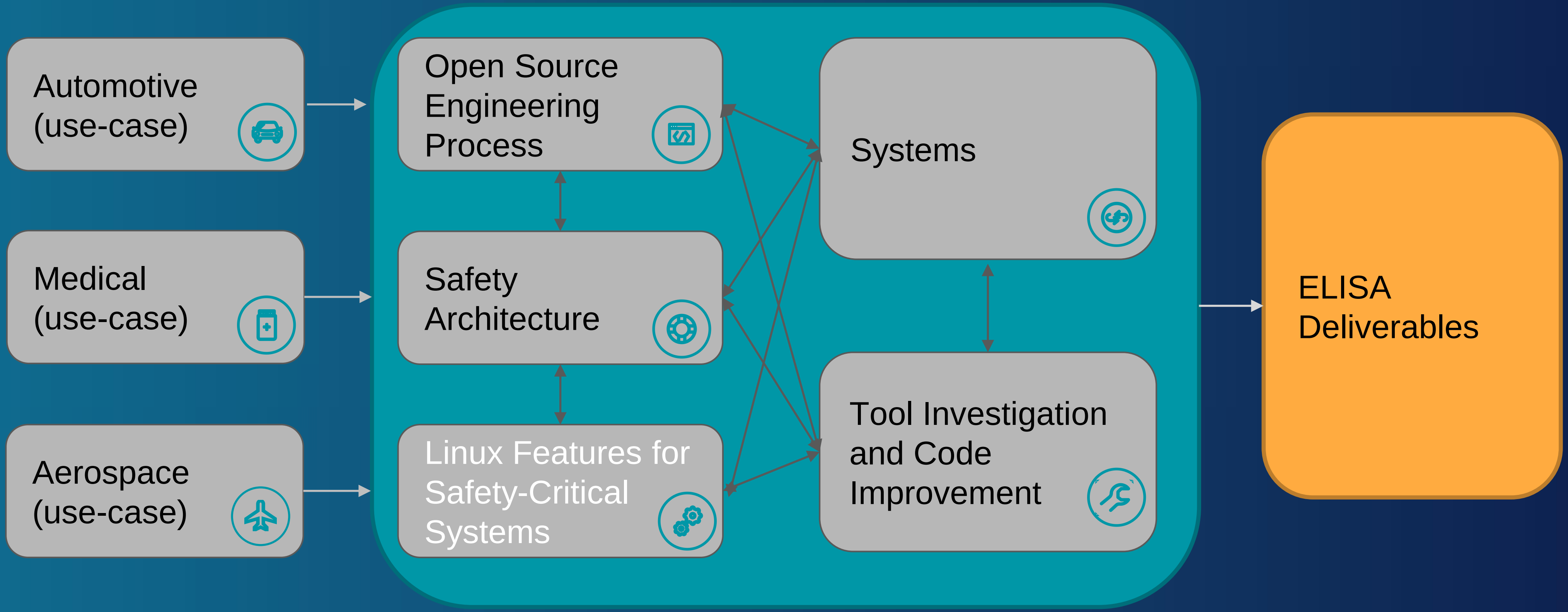
Codethink



Dana Lewis' OpenAPS project: https://youtu.be/kgu-AYSnyZ8

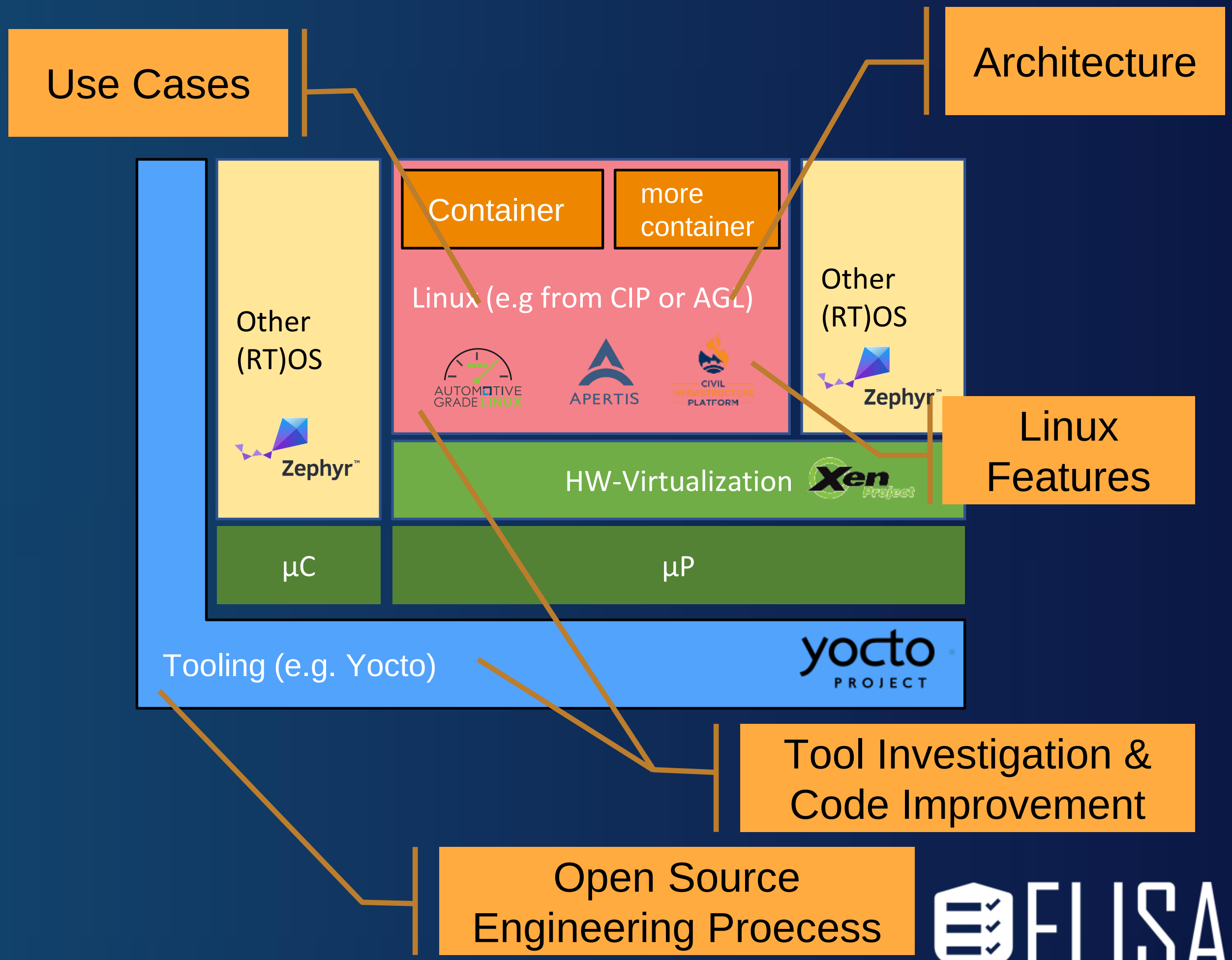
Photo by Hennie Stander on Unsplash

ELISA Working Groups



ELISA Working Groups - Fit in an exemplary system

- **Linux Features, Architecture** and **Code Improvements** should be integrated into the reference system directly.
- **Tools** and **Engineering process** should serve the reproducible product creation.
- **Medical, Automotive, Aerospace** and future WG use cases should be able to strip down the reference system to their use case demands.



ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Interaction with other communities (outside of ELISA)

- Safety-critical open source projects

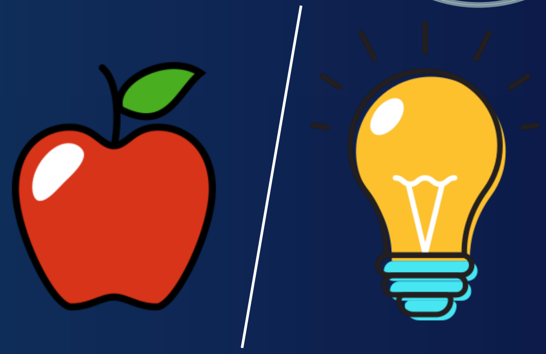
Xen Project
Zephyr™

- OSS project with safety-critical relevance and comparable system architecture considerations

AUTOMOTIVE GRADE LINUX
SOAFEE
SDV™ Eclipse Software Defined Vehicle

- Further interactions

yocto PROJECT
SPDX



"If you have an apple and I have an apple and we exchange these apples then you and I will still each have one apple.
But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas."

— George Bernard Shaw

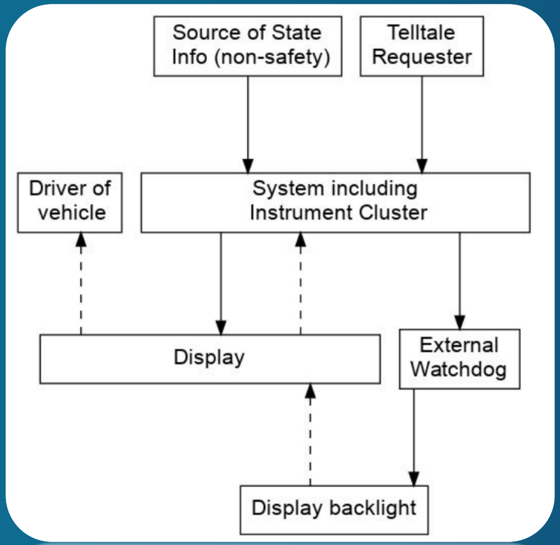
ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Artifacts & Activities

ELISA Working Groups - Deliverables

- Elements / Software — meta-elisa
- Processes — STPA
- Tools — Codechecker, Workload tracing, Call-Tree
- Documentation — GitHub / Gdrive / Blog / Whitepaper

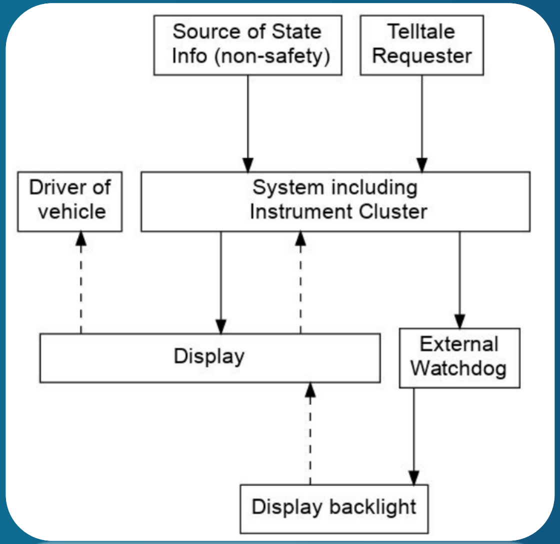
ELISA
ENABLING LINUX IN SAFETY APPLICATIONS



S T P A

System - Theoretic Process Analysis

STPA – Basic Idea

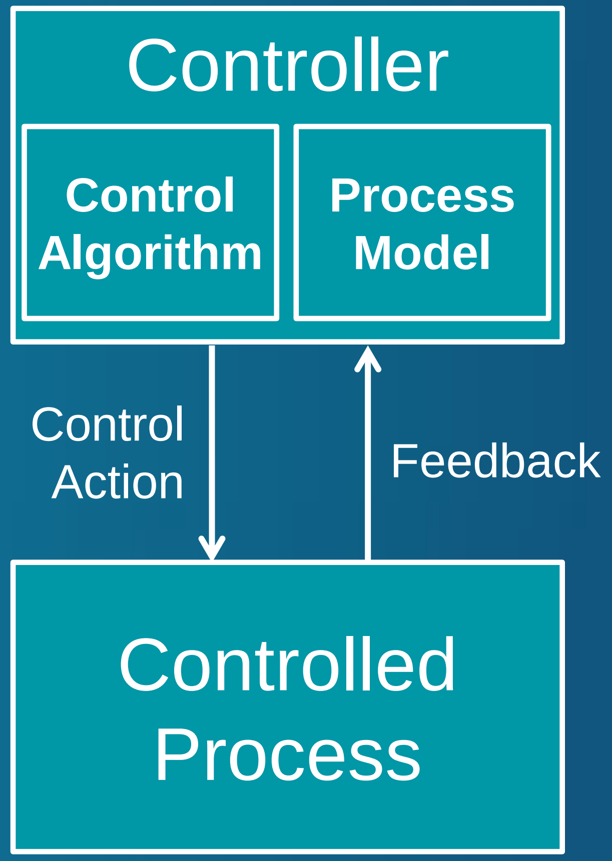


- Relatively new hazard analysis technique
- Very complex systems can be analyzed
- Iterative towards detailed design decisions
- Includes software and human operators
- Provides documentation of system functionality
- Can be easily integrated into (model-based) system engineering process

Reading more about the methodology in the handbook:
https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

STPA – Basic Idea



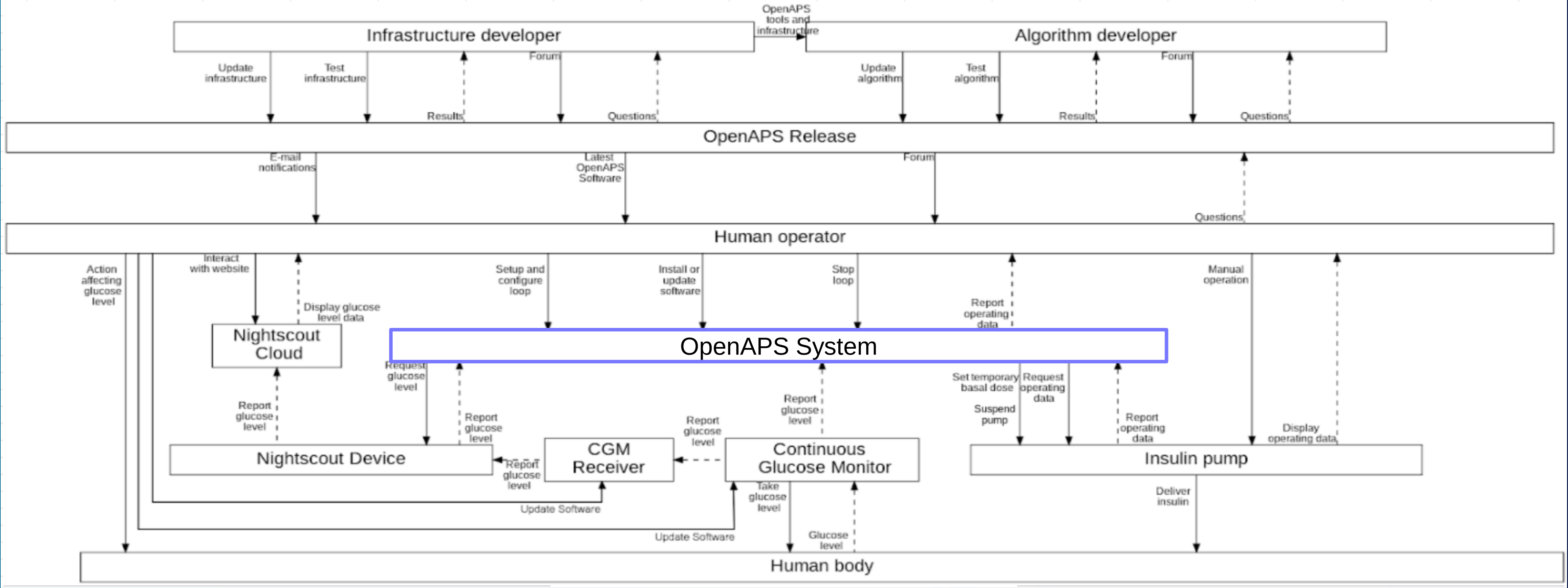
4 key elements

- Controller sends
- Control Action(s) to a
- Controlled Process which provides
- Feedback
- A controlled process can be a controller

Q: What can be unsafe control actions?

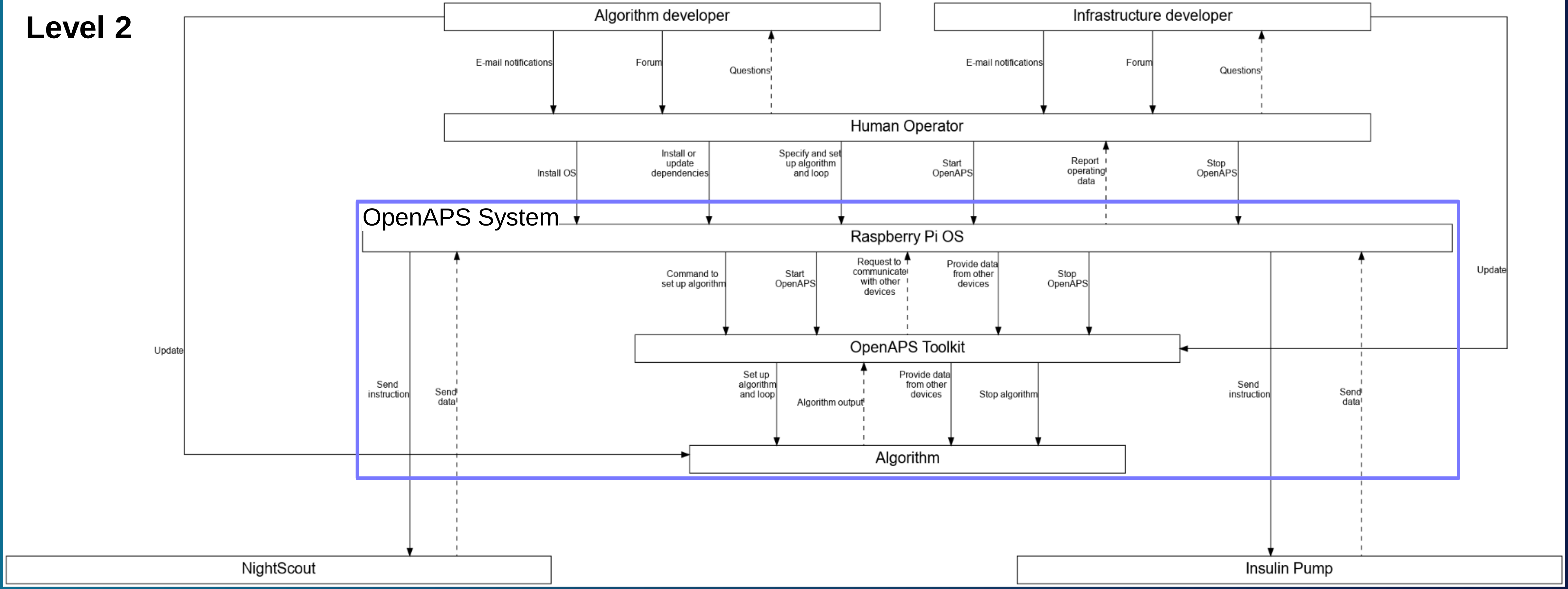
ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

STPA – Example for OpenAPS



ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

STPA – Example for OpenAPS



ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Workload tracing

Main tools used:

- strace
- cscope

Extract information:

- System Call
- Frequency of call
- Involved Subsystem
- System Call Entry Point

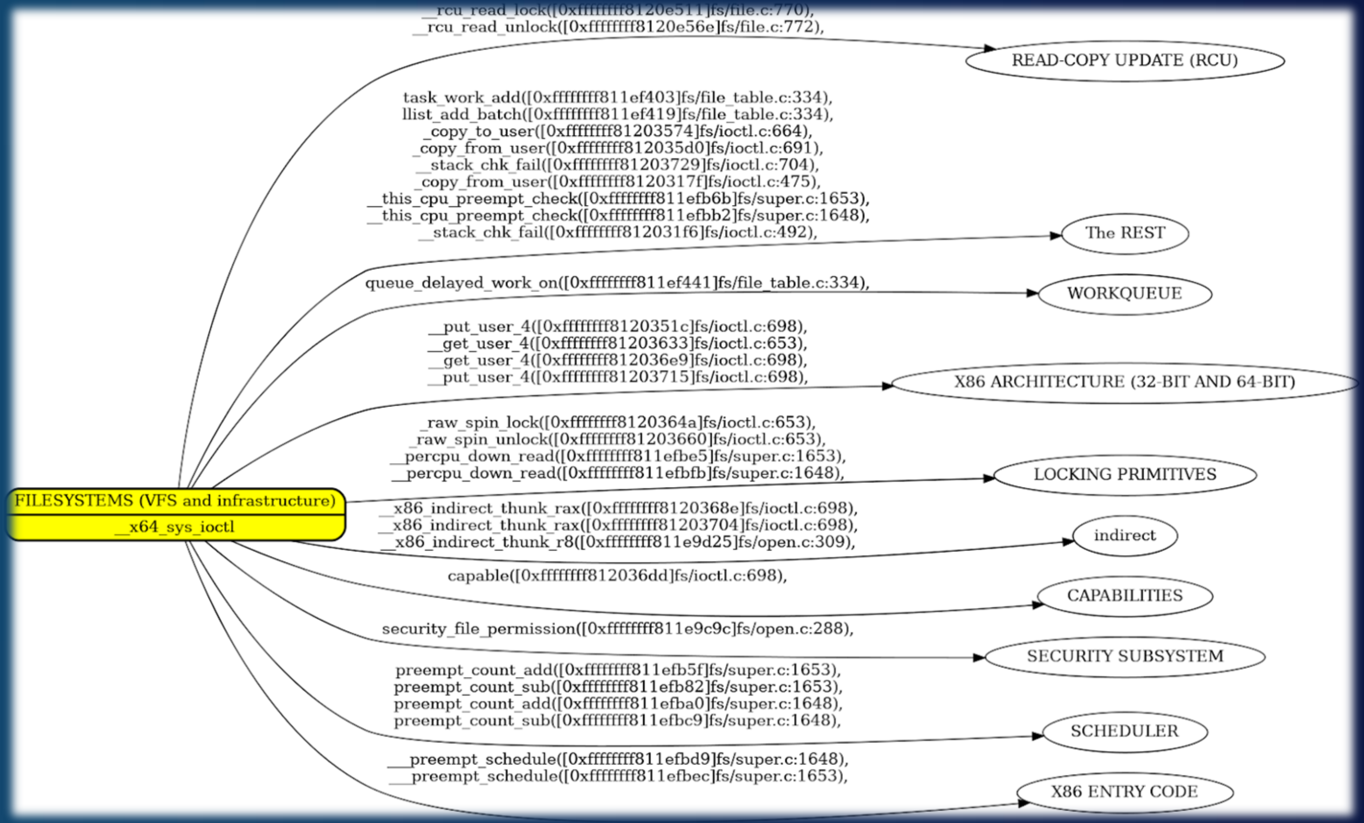
System Call	Frequency	Linux Subsystem	System Call Entry Point (API)
read	3	Filesystem	sys_read()
write	11	Filesystem	sys_write()
close	41	Filesystem	sys_close()
stat	24	Filesystem	sys_stat()
fstat	2	Filesystem	sys_fstat()
pread64	6	Filesystem	sys_pread64()
access	1	Filesystem	sys_access()
pipe	1	Filesystem	sys_pipe()
dup2	24	Filesystem	sys_dup2()
execve	1	Filesystem	sys_execve()
fcntl	26	Filesystem	sys_fcntl()
openat	14	Filesystem	sys_openat()
rt_sigaction	7	Signal	sys_rt_sigaction()
rt_sigreturn	38	Signal	sys_rt_sigreturn()
clone	38	Process Mgmt.	sys_clone()
wait4	44	Time	sys_wait4()
mmap	7	Memory Mgmt.	sys_mmap()
mprotect	3	Memory Mgmt.	sys_mprotect()
munmap	1	Memory Mgmt.	sys_munmap()
brk	3	Memory Mgmt.	sys_brk()
getpid	1	Process Mgmt.	sys_getpid()
getuid	1	Process Mgmt.	sys_getuid()
getgid	1	Process Mgmt.	sys_getgid()
geteuid	2	Process Mgmt.	sys_geteuid()
getegid	1	Process Mgmt.	sys_getegid()
getppid	1	Process Mgmt.	sys_getppid()
arch_prctl	2	Process Mgmt.	sys_arch_prctl()

ENABLING LINUX IN SAFETY APPLICATIONS

https://github.com/elisa-tech/ELISA-White-Papers/blob/master/Processes/Discovering_Linux_kernel_subsystems_used_by_a_workload.md

Call Tree Tool

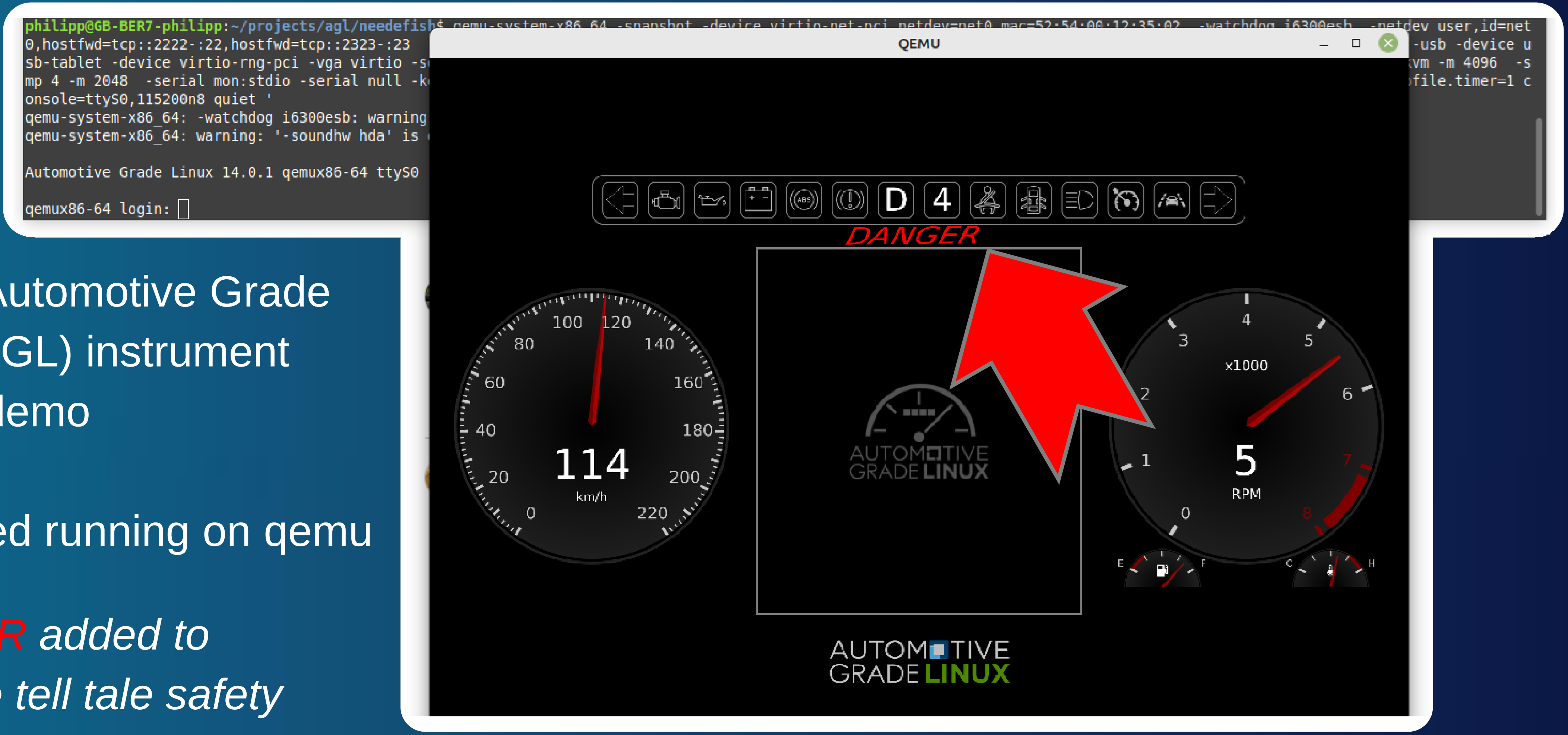
- Supports the analysis on code/kernel level
- Graphical representation of source code
- Provides insights about the Kernel construction



https://github.com/elisa-tech/Safety_Architecture_WG/tree/main/call_tree_tool

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

meta-elisa: Instrument cluster example derived from AGL



- Reuse Automotive Grade Linux (AGL) instrument cluster demo
- QT based running on qemu
- *DANGER added to illustrate tell tale safety monitoring*

ENABLING LINUX IN SAFETY APPLICATIONS

https://github.com/elisa-tech/meta-elisa

External Watchdog

- The challenge-response watchdog serves as the “safety net” for the safety-critical workload
- The concept is widely used in Automotive and other industrial applications
- It can be used as an iterative approach to assign more safety-critical functionality to Linux

With a proper system design the watchdog will never need to trigger the “safe state”.

[Standardized E-Gas Monitoring Concept for Gasoline and Diesel Engine Control Units]

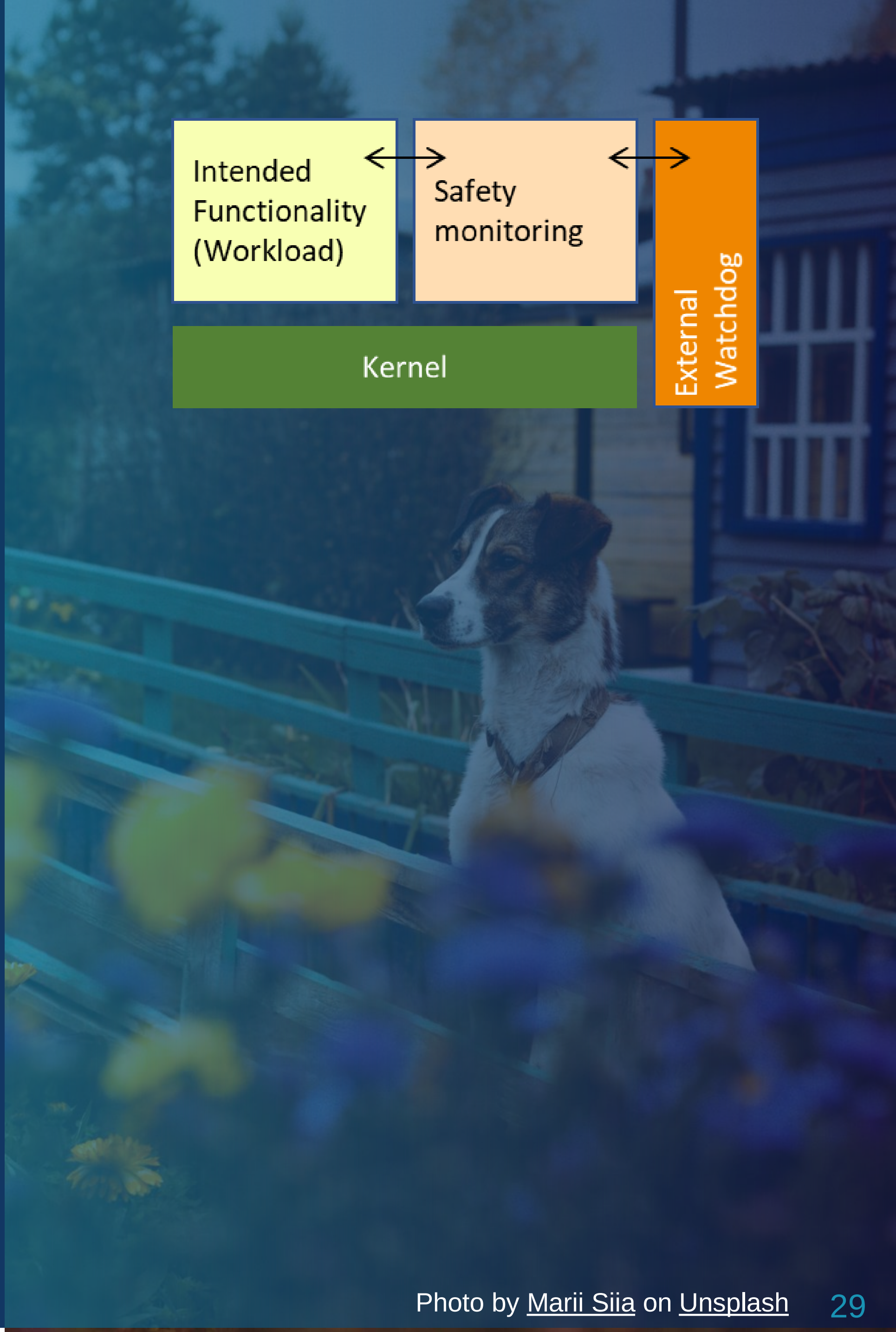


Photo by Marii Siia on Unsplash

Use case recording on the event side

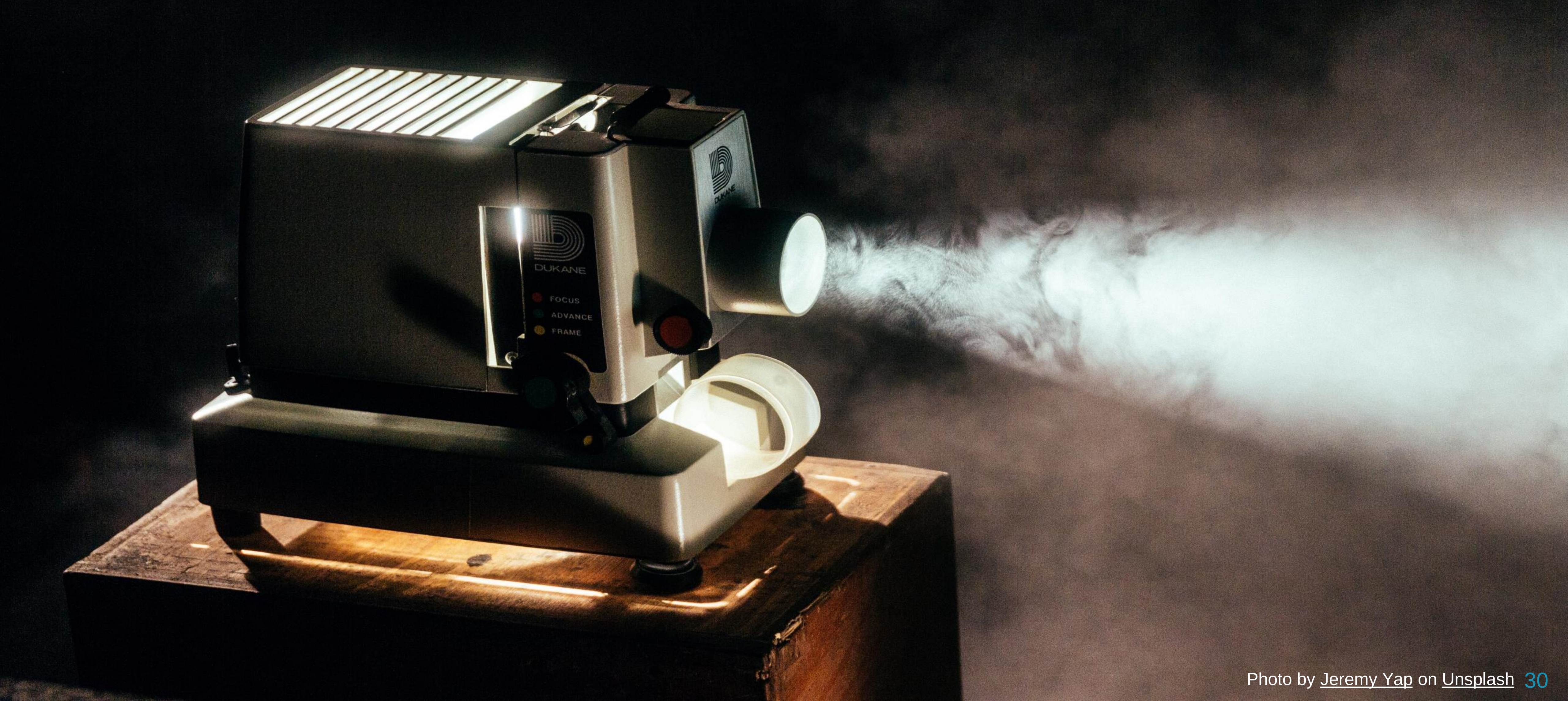


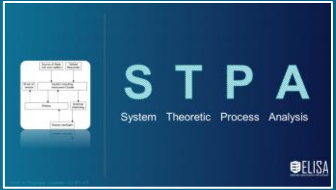
Photo by Jeremy Yap on Unsplash

meta-elisa: Various starting points provided

- Plain and native from source
  https://github.com/elisa-tech/meta-elisa
- Using docker container
  https://github.com/elisa-tech/wg-automotive/tree/master/Docker_container
- With cached build using SSTATE
  modify "`conf/local.conf`" after the "`source`" command before the "`bitbake`" command
- Download binaries directly from build server
  https://gitlab.com/elisa-tech/meta-elisa-ci

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Next steps



STPA

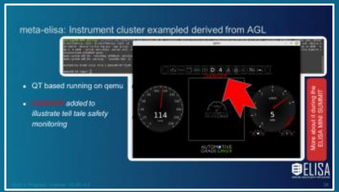
- Align implementation and design using workload tracing and Linux analysis

Call Tree

- Extend call tree with additional tools. PR for a tool called “ks-nav” is already open

Workload tracing

- Use the workload tracing from medical devices for the automotive use case
- Add other tracing tools like SystemTap



meta-elisa

- Bring the qemu showcase on real hardware using the work of the systems WG
- Improve the monitoring and checking apps

And more: System SBOM generation, Kernel configuration & image size trim down, RT docu review, cluster demo on complex system architecture, more docu…

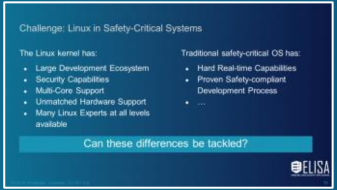
ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

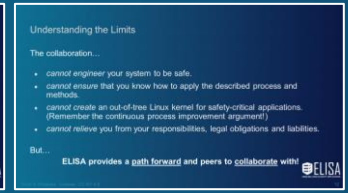
Conclusion

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Conclusion

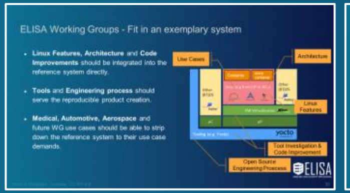
- Challenges



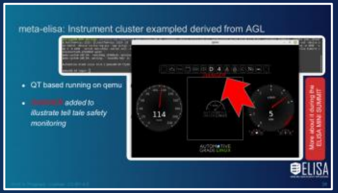


- Goals and technical strategy

- Different work groups & their interaction



- Contributions & Deliverables

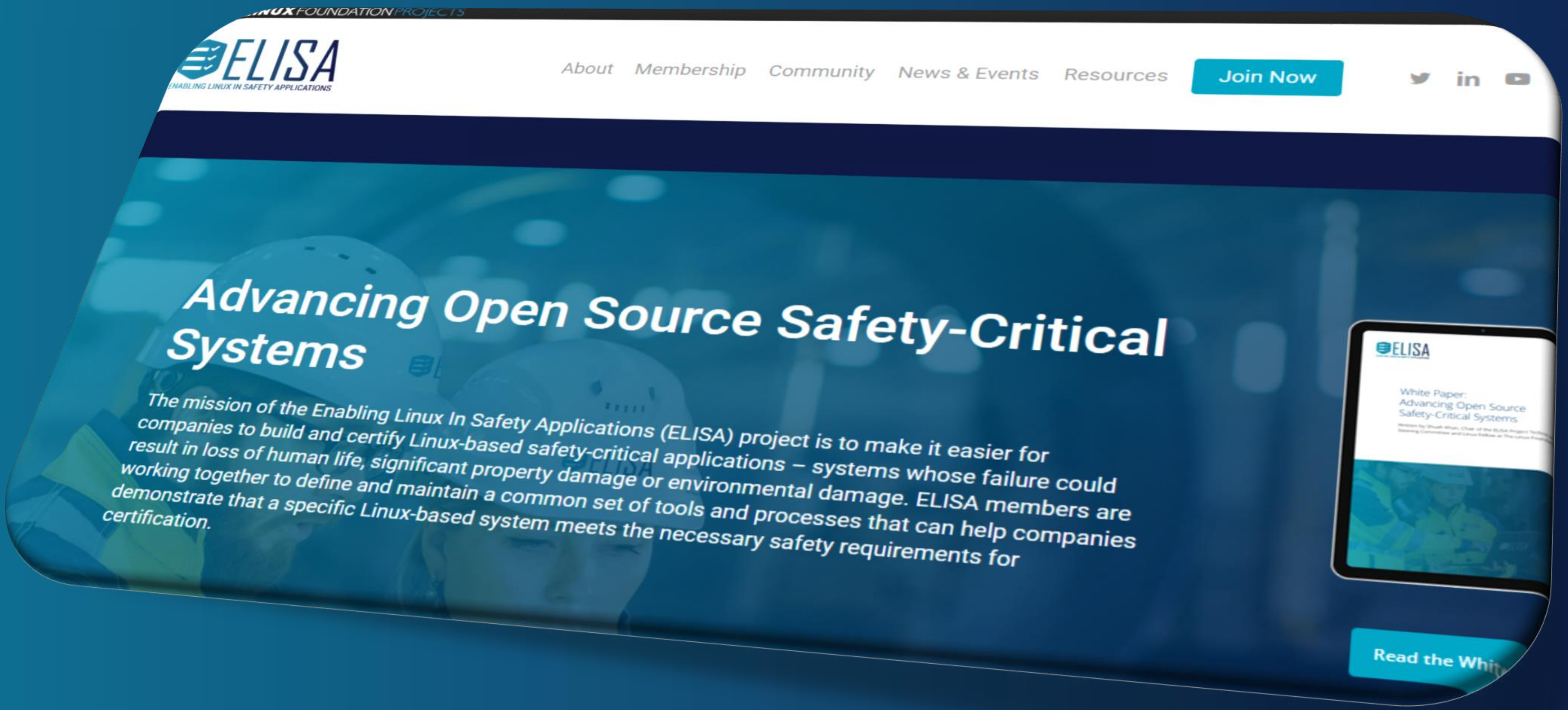


- Used methodologies and tools

- Current status & what comes next

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Questions? info@elisa.tech



ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Getting involved...

- Join main technical and weekly calls of interest:
  - Main Technical List: devel@lists.elisa.tech
  - Safety Architecture Workgroup: safety-architecture@lists.elisa.tech
  - Open-Source Engineering Process WG osep@lists.elisa.tech
  - Linux Features for Safety-Critical Systems WG: linux-features@lists.elisa.tech
  - Medical Devices Workgroup: medical-devices@lists.elisa.tech
  - Automotive Workgroup: automotive@lists.elisa.tech
  - Systems Workgroup: systems@lists.elisa.tech
  - *(Full list at: https://lists.elisa.tech/g/linux-features/subgroups)*

- Contribute content, review materials and add your comments to:
  - ELISA Technical Community Google Drive: https://drive.google.com/open?id=1Y6Uwqt5VEDEZjpRe0CBCIibdtXPgDwlG
  - ELISA github repository: https://github.com/elisa-tech/workgroups
  - ELISA github issue tracker: https://github.com/elisa-tech/workgroups/issues
  - “Final location” for (Architecture/Process/…) Documentation on kernel: https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/Documentation

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Advancing Open Source Safety-Critical Systems

Elements ▪ Processes ▪ Tools ▪ Documentation

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS