

Automotive WG Update

Philipp Ahmann, Robert Bosch GmbH

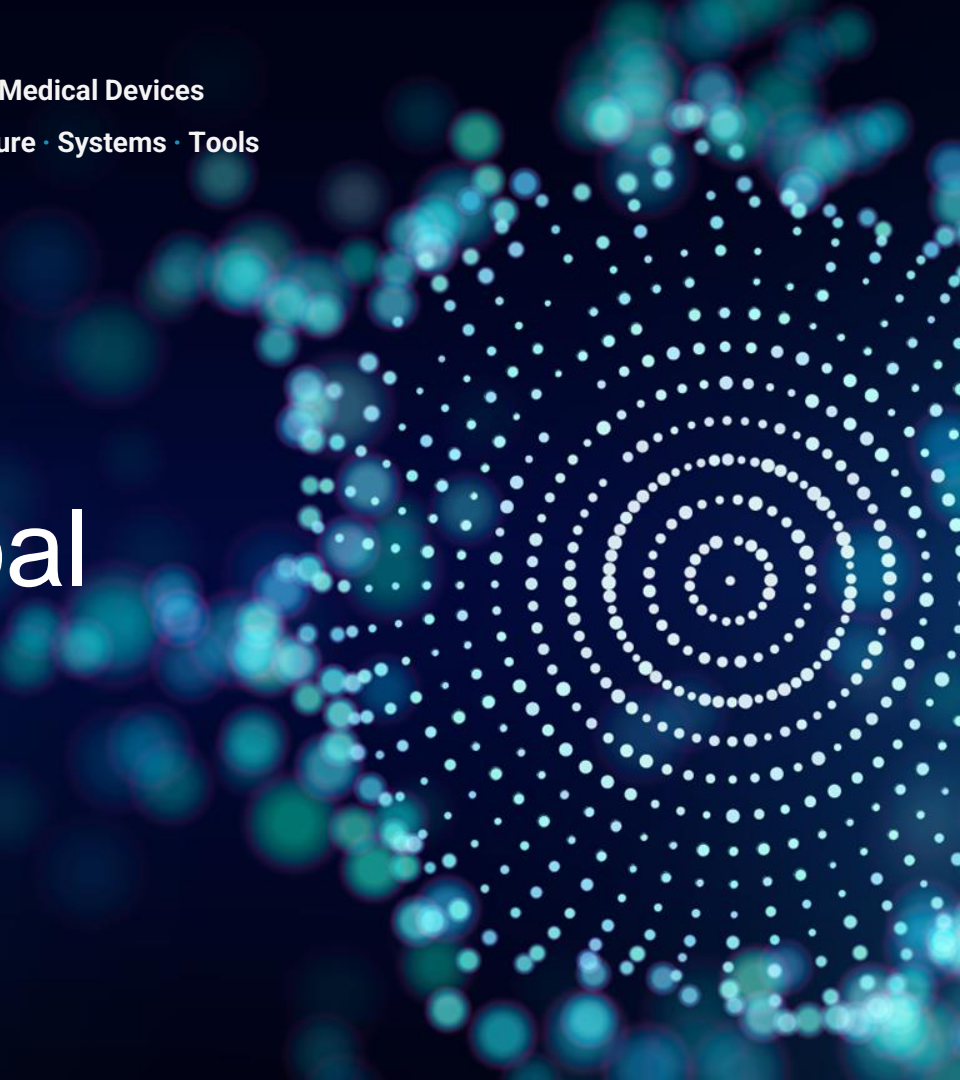


Aerospace · Automotive · Linux Features
Medical Devices · OS Engineering Process
Safety Architecture · Systems · Tools

Topics

- Working group goal & introduction
- Milestones & achievements in 2023
- Challenges and fails
- Current focus / activities
- Plans for 2024 & collaboration opportunities

Working Group Goal & Introduction



Working Group Goal

“Discuss the conditions and prerequisites the automotive sector needs to integrate Linux into a safety critical system.

We focus on actual use cases from the Automotive domain to derive the technical requirements to the kernel and the development process.”



Aerospace · Automotive · Linux Features · Medical Devices
OS Engineering Process · Safety Architecture · Systems · Tools

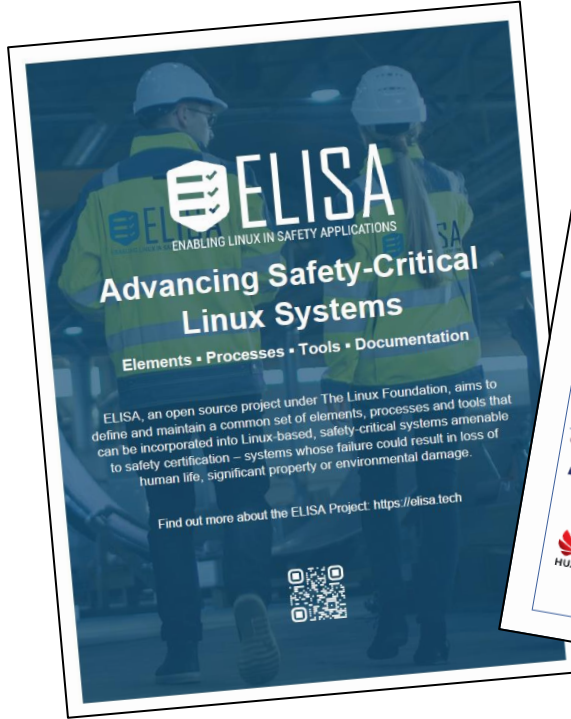
Milestones & Achievements



Milestones & Achievements

- One pager for Automotive Grade Linux (AGL) at CES and embedded world...
- Panel with Eclipse Software Defined Vehicle at Exida Symposium
- Reproducible CI Setup of meta-elisa (AGL enhanced cluster demo)
 - Daily tested builds with downloadable images
 - Including SBOM generation for yocto parts
- Strong participation of Automotive companies during ELISA Workshops
- Many automotive centric material created by members and affiliates

ELISA One Pager




ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Advancing Safety-Critical Linux Systems

Elements • Processes • Tools • Documentation

ELISA, an open source project under The Linux Foundation, aims 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 – systems whose failure could result in loss of human life, significant property or environmental damage.

Find out more about the ELISA Project: <https://elisa.tech>



ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

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

Horizontal Working Groups

- Architecture
- Tool Investigation & Code Improvement
- Linux Features
- Open Source Engineering Process
- Systems

Exemplary reference architecture from ELISA's Systems WG



Use Case Verticals

- Automotive
- Medical
- Aerospace

Automotive use case: Instrument cluster warning signs (tell tales)



Medical devices use case: Open Artificial Pancreas System (OpenAPS)



ELISA contributes building blocks needed to ease the path for Linux-based safety-critical systems. These include:

- System & kernel analysis processes & tools
- Argumentation for a safety integrity standard
- Equivalent development process description
- Explicit Linux features, to enhance system safety
- Use case based reference systems

An essential element is the usage of an external challenge-response watchdog, a concept used widely in Automotive and other industries. The critical workload in the Linux system.

Premier Members

- BOEING
- Red Hat

General Members

- AISIN
- arm
- 斑马智行 (Powered by AISIN)
- BOSCH
- Geelythink
- ES
- Elektrobit
- 地平线 (Horizon Robotics)
- HUAWEI
- LINUTRONIX
- SAIC
- 上汽集团 (SAIC MOTOR)
- SUZUKI
- WINDRVR
- SUSE

Associate Members

- AUTOMOTIVE GRADE LINUX
- OTH

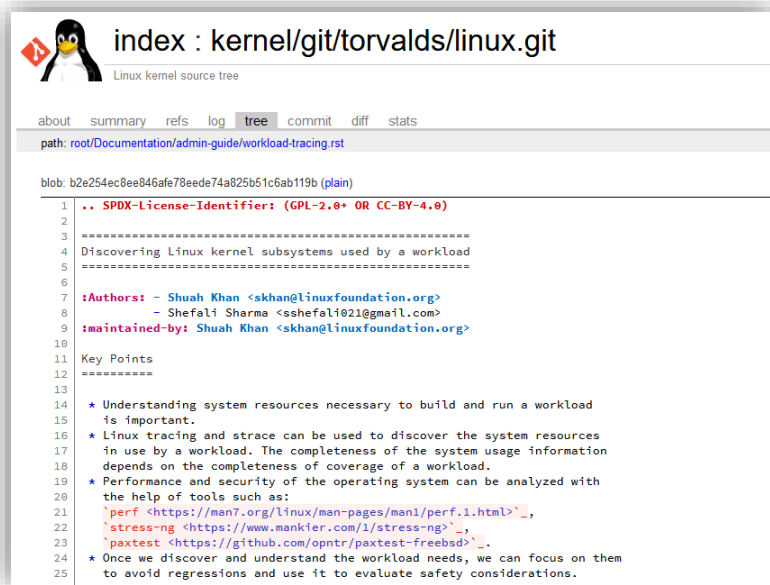
Industry Support

- CIVIL AVIATION AUTHORITY PLATFORM
- OSADL
- UL

Workload Tracing Packages Added to the CI

- Documentation upstream available:

linux.git/tree/Documentation/admin-guide/workload-tracing.rst



The screenshot shows the index page for the kernel documentation file `Documentation/admin-guide/workload-tracing.rst`. The page title is "index : kernel/git/torvalds/linux.git". The content includes a license identifier: `.. SPDX-License-Identifier: (GPL-2.0+ OR CC-BY-4.0)`. It also lists authors: Shuah Khan and Shefali Sharma, and a maintainer: Shuah Khan. Key points are listed, such as "Understanding system resources necessary to build and run a workload is important." and "Linux tracing and strace can be used to discover the system resources in use by a workload." Tools mentioned include `perf`, `stress-ng`, and `paxtest`.

```
root@gemux86-64:~# strace -c stress-ng --netdev 1 -t 60 --metrics
stress-ng: info: [395] setting to a 60 second run per stressor
stress-ng: info: [395] dispatching hogs: 1 netdev
stress-ng: info: [395] successful run completed in 60.01s (1 min,
0.01 secs)
stress-ng: info: [395] stressor bogo ops real time usr
time sys time bogo ops/s bogo ops/s CPU used per
stress-ng: info: [395] (secs) (secs) (secs) (secs)
(secs) (real time) (usr+sys time) instance (s)
stress-ng: info: [395] netdev 13566190 60.00 8.17 48.40
226102.70 229812.44 94.28
% time seconds usecs/call calls errors syscall
-----
99.97 48.405401 24202700 2 1 wait4
0.01 0.004856 539 9 write
0.01 0.004529 40 111 9 openat
0.00 0.001011 9 105 close
0.00 0.000920 10 92 read
0.00 0.000863 3 2 connect
0.00 0.000800 3 0 mmap
0.00 0.000788 3 0 mmap64
0.00 0.000773 173 6 22
0.00 0.000138 23 6
0.00 0.000123 41 3 sysinfo
0.00 0.000121 8 14 flock
0.00 0.000118 39 3 socket
0.00 0.000117 29 4 2 access
0.00 0.000102 11 9 getppid
0.00 0.000051 10 5 prlimit64
0.00 0.000049 16 3 sendto
0.00 0.000046 46 1 getppid
0.00 0.000026 26 1 getrusage
0.00 0.000026 26 1 stats
0.00 0.000017 5 3 geteuid
0.00 0.000015 5 3 brk
0.00 0.000013 6 2 getuid
0.00 0.000012 6 2 lseek
0.00 0.000012 12 1 1 rt_sigreturn
0.00 0.000011 11 1 getcwd
0.00 0.000010 2 4 pread64
0.00 0.000008 8 1 getrandom
0.00 0.000007 7 1 sigaltstack
0.00 0.000005 5 1 uname
0.00 0.000002 2 1 alarm
0.00 0.000002 2 1 1 setpgid
0.00 0.000002 1 2 1 arch_prctl
0.00 0.000002 2 1 set_tid_address
0.00 0.000001 1 1 set_robust_list
0.00 0.000001 1 1 req
0.00 0.000000 0 1 execve
-----
100.00 48.419806 97228 498 25 total
root@gemux86-64:~#
```

strace
strace-log-merge stress
stress-ng
strings

Challenges & Fails

- Automotive business remains a very conservative and intellectual property driven business
- Setting a use case with the right balance between attractiveness and complexity.
 - Something else then telltale (warning signs) is needed.

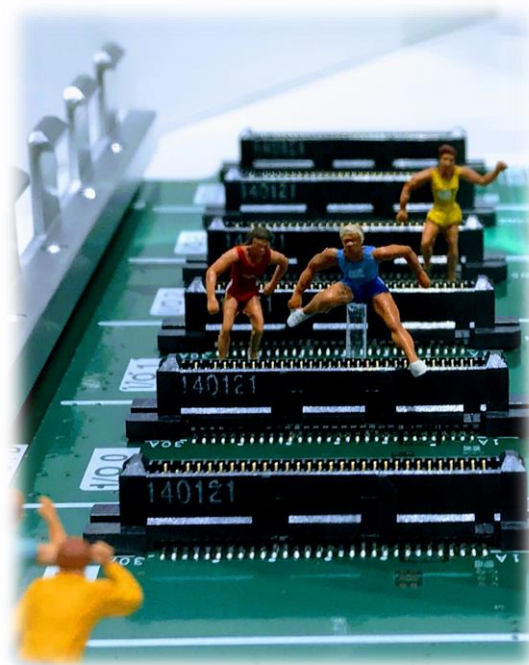


Photo by [John Cameron](#) on [Unsplash](#)



Aerospace · Automotive · Linux Features · Medical Devices
OS Engineering Process · Safety Architecture · Systems · Tools

meta-elisa

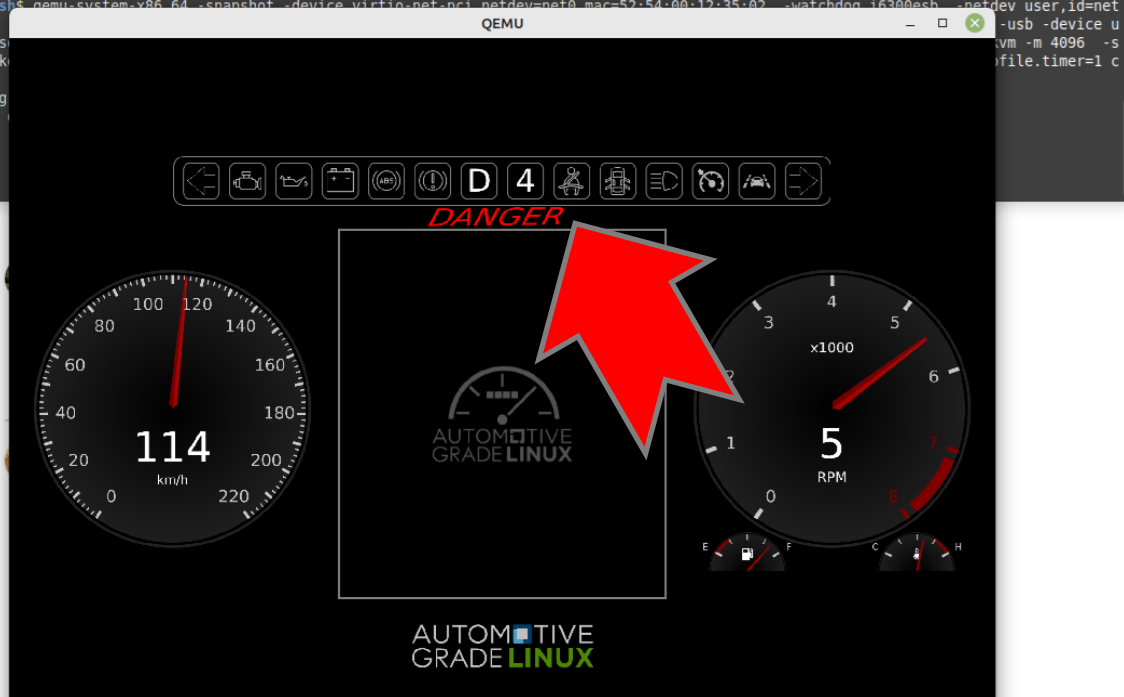
CI Details



AGL Instrument Cluster Enhancements

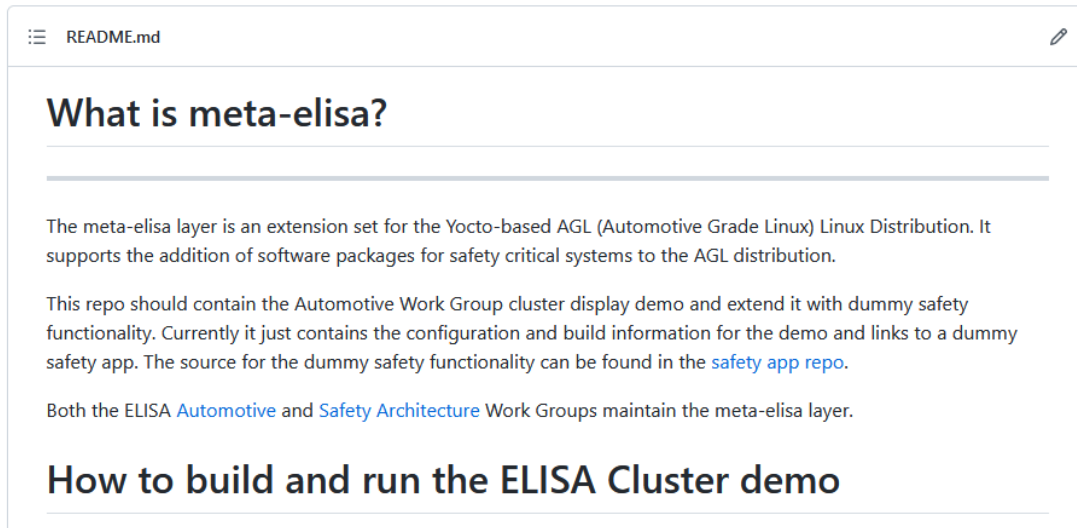
```
philipp@66-BER7-philipp:~/projects/agl/needefish$ qemu-system-x86_64 -snapshot -device virtio-net-pci,netdev=net0,mac=52:54:00:12:35:02,watchdog,16300esb,netdev user,id=net0,hostfwd=tcp::2222-:22,hostfwd=tcp::2323-:23 -usb -device usb-tablet -device virtio-rng-pci -vga virtio -smp 4 -m 2048 -serial mon:stdio -serial null -kvm -onion -onion console=ttyS0,115200n8 quiet 'qemu-system-x86_64: -watchdog i6300esb: warning: qemu-system-x86_64: warning: '-soundhw hda' is not availableAutomotive Grade Linux 14.0.1 qemux86-64 ttyS0qemux86-64 login: []
```

- QT based running on qemu
- **DANGER** added to illustrate tell tale safety monitoring



Instrument Cluster CI

- Sources are fetched from meta-elisa
<https://github.com/elisa-tech/meta-elisa>



☰ README.md ✎

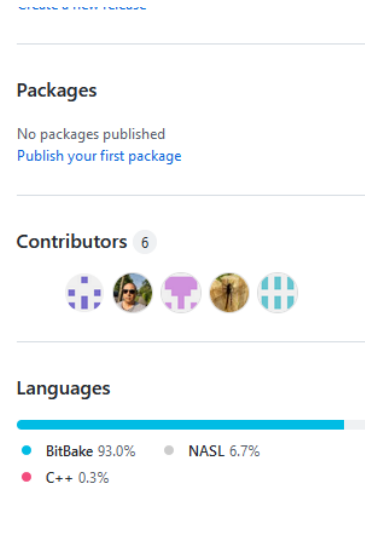
What is meta-elisa?

The meta-elisa layer is an extension set for the Yocto-based AGL (Automotive Grade Linux) Linux Distribution. It supports the addition of software packages for safety critical systems to the AGL distribution.

This repo should contain the Automotive Work Group cluster display demo and extend it with dummy safety functionality. Currently it just contains the configuration and build information for the demo and links to a dummy safety app. The source for the dummy safety functionality can be found in the [safety app repo](#).

Both the ELISA [Automotive](#) and [Safety Architecture](#) Work Groups maintain the meta-elisa layer.

How to build and run the ELISA Cluster demo




CREATE NEW PACKAGE

Packages


No packages published
[Publish your first package](#)

Contributors

6



Languages











Language	Percentage
BitBake	93.0%
NASL	6.7%
C++	0.3%

meta-elisa: CI Enablement

elisa-tech / meta-elisa-ci Jobs

All 1,000+ Finished

Filter jobs

Status	Job	Pipeline	Coverage
 Passed 00:04:10 19 hours ago	#5935140115: push_package main d1dab59a	#1139219660 created by  Stage: package	
 Passed 00:01:11 19 hours ago	#5935140110: qemu-boot main d1dab59a elisa	#1139219660 created by  Stage: qemu	
 Passed 00:09:54 19 hours ago	#5935140104: meta-elisa-build main d1dab59a elisa	#1139219660 created by  Stage: build	
 Passed 00:04:00 1 day ago	#5927478368: push_package main d1dab59a	#1137975087 created by  Stage: package	

Runs daily

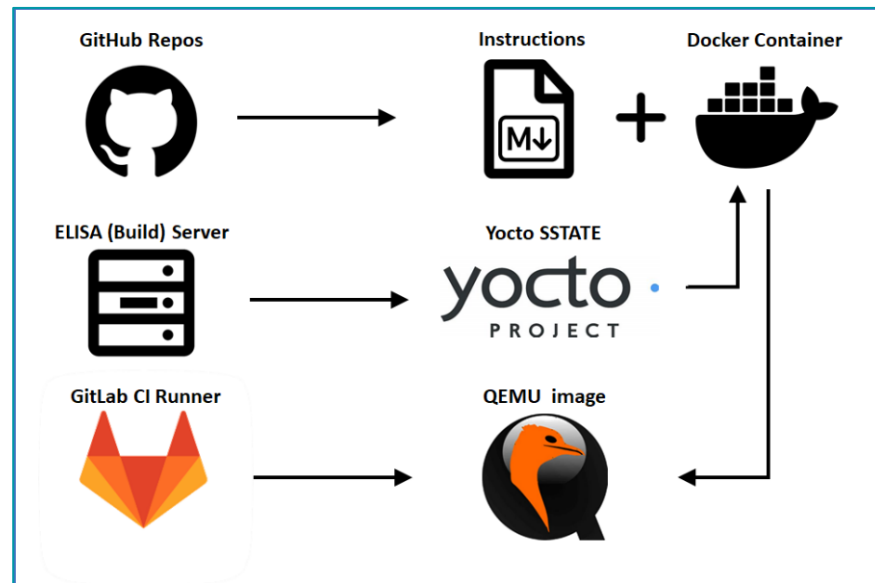


Artifacts download of recent images

<https://gitlab.com/elisa-tech/meta-elisa-ci>

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>



AGL Cluster Demo SBOM Available

- Enablement by adding `INHERIT += "create-spdx"` to your build
- `"SPDX_PRETTY"` for human readable form

<https://gitlab.com/elisa-tech/meta-elisa-ci/-/pipelines>

AGL Cluster Demo SBOM Available

Why GitLab Pricing Contact Sales Explore

elisa-tech / meta-elisa-ci / Pipelines

All 462 Finished Branches Tags

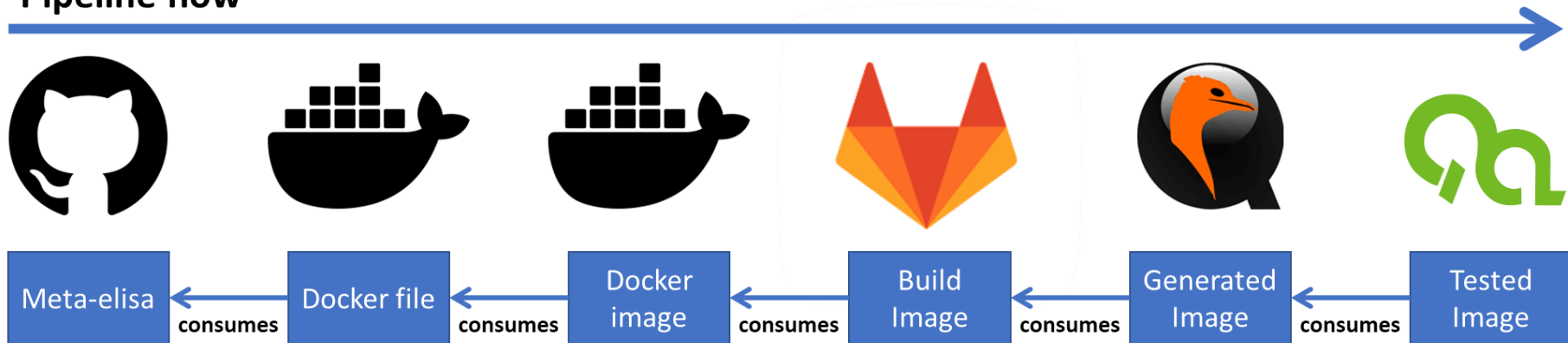
Filter pipelines

Status	Pipeline	Created by	Stages
✓ Passed ⌚ 00:14:53 📅 12 hours ago	Generate sbom #1137975087 main d1dab59a scheduled latest		✓ ✓ ✓
✓ Passed ⌚ 00:15:06 📅 1 day ago	Generate sbom #1137480094 main d1dab59a scheduled latest		✓ ✓ ✓

<https://gitlab.com/elisa-tech/meta-elisa-ci/-/pipelines>

meta-elisa: Pipeline Dependencies

Pipeline flow



Full description in the blog

<https://elisa.tech/blog/2023/04/05/elisa-ci-enablement-automation-tools-for-easier-collaboration/>



Aerospace · Automotive · Linux Features · Medical Devices
OS Engineering Process · Safety Architecture · Systems · Tools

Current Focus / Activities

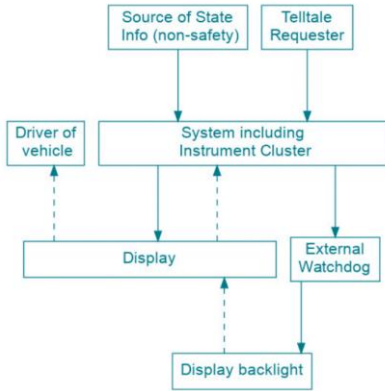


Current Focus / Activities

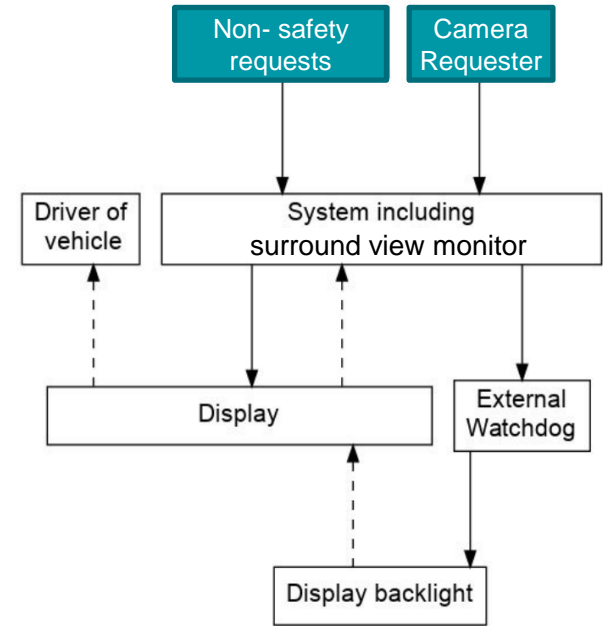
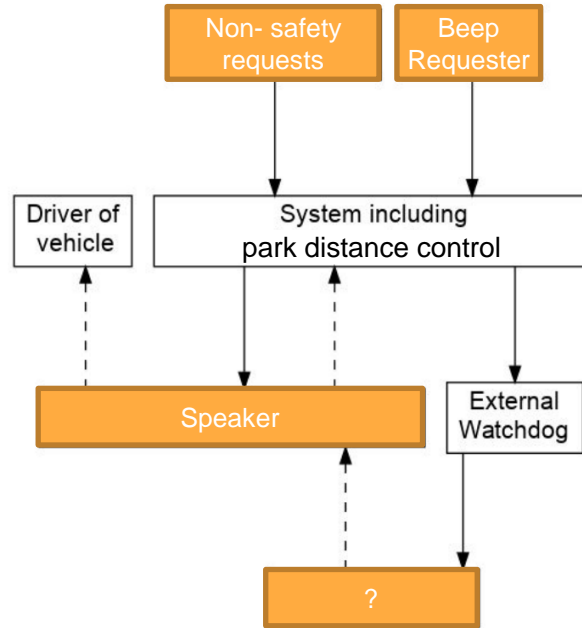
- Interaction with other communities like Eclipse SDV, SOAFEE, AGL, ...
- Looking for a new use case after the telltale use case where the analysis work can be applied to.

Similarity With Other Use Cases

Basic challenges representative for more complex use cases



Use the tell tale use case work to derive your own use case.





Aerospace · Automotive · Linux Features · Medical Devices

OS Engineering Process · Safety Architecture · Systems · Tools

Plans for 2024 & Collaboration Opportunities



Plans for 2024 & Collaboration Opportunities

- Acquire a new use case driven by an OEM or Tier1.
- Support other communities who miss safety focus/knowledge.
- Reach out to AGL, Eclipse SDV, SOAFEE, COVESA...

Join our journey with a simple mail on the mailing list: <https://lists.elisa.tech/g/automotive>

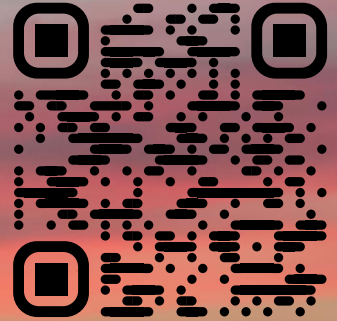
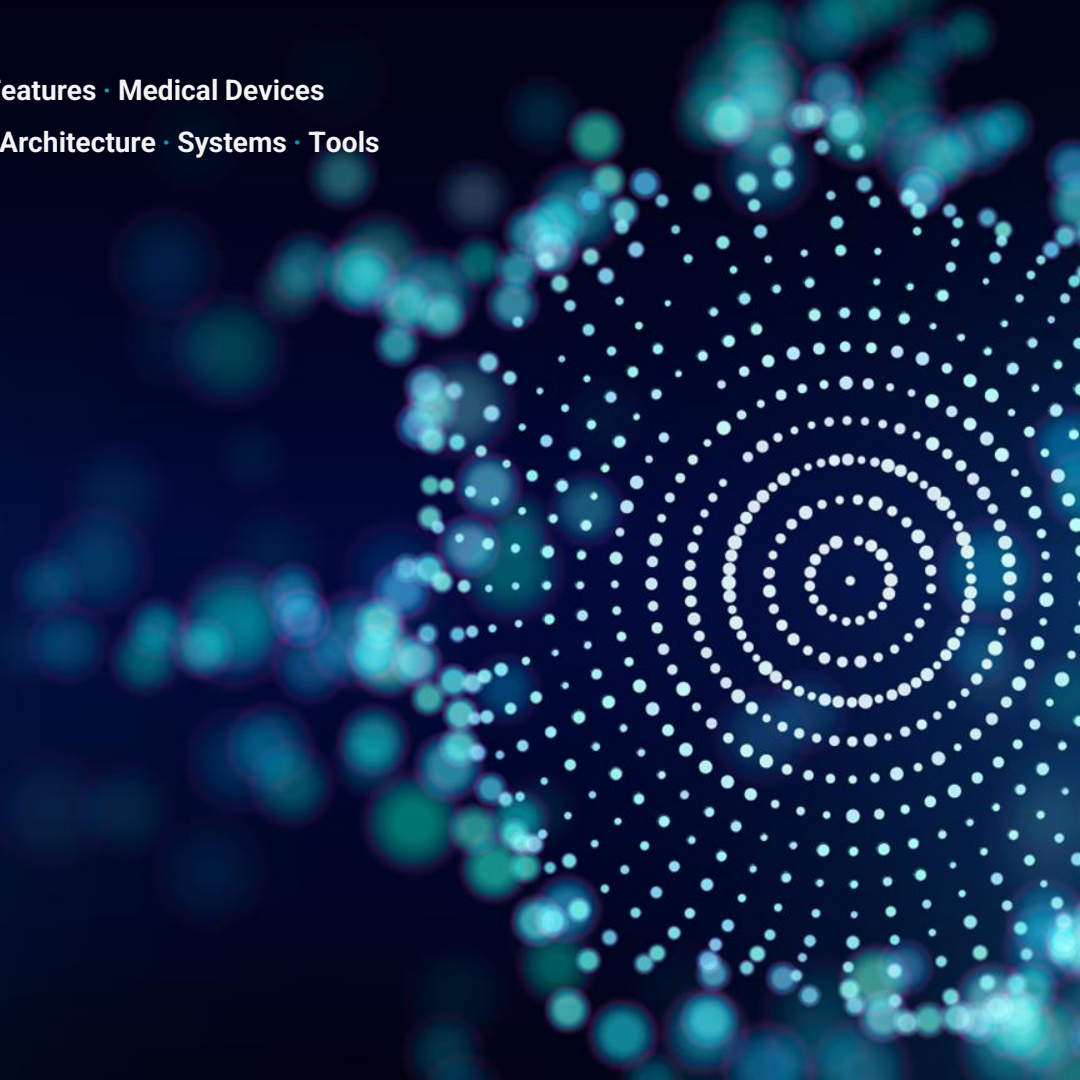


Photo by Javier Allegue Barros on Unsplash



Aerospace · Automotive · Linux Features · Medical Devices
OS Engineering Process · Safety Architecture · Systems · Tools

Thank you



JOIN THE COMMUNITY

ELISA members are defining and maintaining a common set of elements, processes and tools that can be incorporated into specific Linux-based, safety-critical systems amenable to safety certification. ELISA is also working with certification authorities and standardization bodies in multiple industries to establish how Linux can be used as a component in safety-critical systems.

Join us to expand the use of Linux across new industries including healthcare, energy, transportation, and manufacturing. Learn more today to participate and support ELISA.



[Join mailing lists](#)



[Participate in meetings](#)



[Contribute to documentations](#)



[Get involved in WGs](#)



[Collaborate at Workshops](#)