CRITICAL SOFTWARE SUMMIT @ THE LINUX FOUNDATION OPEN SOURCE SUMMIT EUROPE

Cross Industry Demands and Collaboration Opportunities in OS for Safety Critical Systems

Philipp Ahmann, ETAS GmbH (BOSCH)
Olivier Charrier, Wind River

Agenda

- Various Safety Integrity Standards
- Similarity and differences for standards.
- Resulting challenges for OSS
- ELISA Project introduction

Part1: Overview & Introduction

- ELISA Deliverables
- ELISA Systems WG
- Education
- Role of documentation

Part2: Initiatives and way forward

ELISA
Enabling Linux in Safety Applications

whoami - Philipp Ahmann

Sr. OSS Community Manager

Chair of the Technical Steering Committee
Lead of the Systems Working Group

Member of the Inaugural Advisory Board

OSS enthusiast and promoter

whoami – Olivier Charrier

WNDRVR — Principal Technologist – Functional Safety

ELISA Enabling Linux in Safety Applications — ELISA Ambassador

SAE / ARINC — Core Member of the APEX Subcommittee

OSS enthusiast

ELISA Enabling Linux in Safety Applications

Part1: Overview & Introduction

Why Open Source Software

- Open Source for prototyping -> A way to certification is interesting.
- Innovation is done with open source as a first step.

Enables addressing industry constraints:

- Consolidation of control units
- Reduction of SWaP & Cabling (Space, weight, power & cabling) in the vehicle.
    - Time & Space partitioning

ELISA
Enabling Linux in Safety Applications

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:

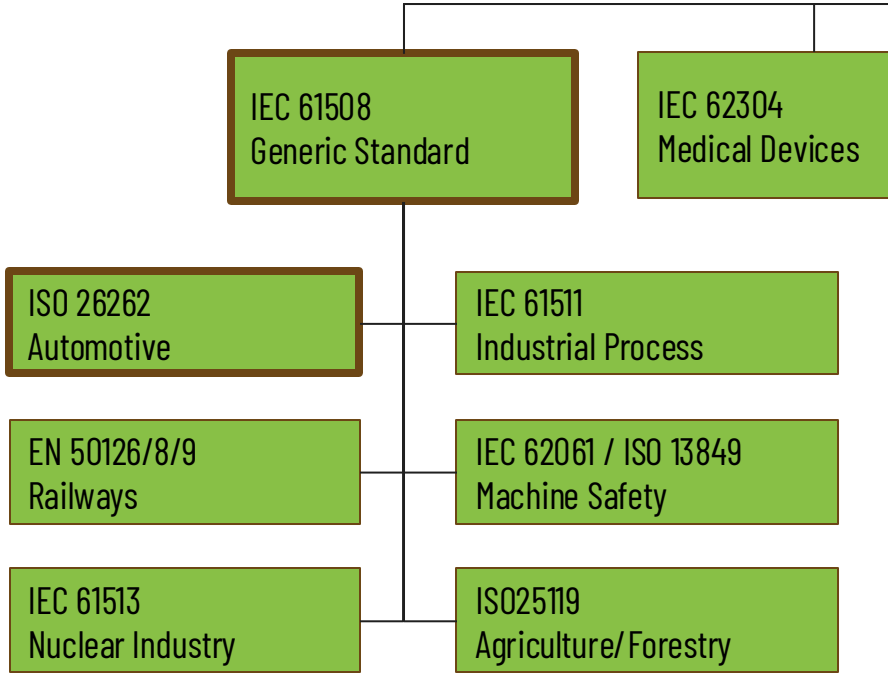
That the software:

- 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.

Credits to Nicole Pappler for the content of this slide.

Samples of safety (integrity) standards



- Standards represent industry best practice
  and recommend to use these practices
- Share similar demands...
  requirements, documentation, testing
- Rigor on measures/demands may differ
- All system parts need to be known, tested and managed

ELISA
Enabling Linux in Safety Applications

Credits to Kate Stewart for the base of this slide.

Similarity and differences for standards.

- Similarities
  - Requirements based standards
  - Traceability has to be demonstrated between Requirements, Design and Tests
  - Failure modes have to be identified and either mitigated or documented

- Differences
  - Management of mixed criticality (Freedom of Interference versus Partitioning)
  - Tooling Qualification (need for a Qualified compiler or not, for example)

Community challenges for OSS projects introduced by standards

- How to map OSS project lifecycle and output to safety standards
  - Development process: Requirements, traceability, v-model,...
- Decision making in OSS project does not include safety standard concerns
  - Different leadership models, roles, restrictions, lifecycle
  - Less influence on maintainers (e.g. forcing into deadline does not work!)
  - but quality is important on both sides
- Liability of a community? *(while commercial provider may be liable – insurance)*
- Harder to train/direct developers
- Document the safe usage of OSS

ELISA
Enabling Linux in Safety Applications

Some OSS projects addressing functional safety gaps and concerns

Linux:

RTOS:

Virtualization/Hypervisor:

ELISA Project

- Enabling **Safety-critical applications** with **Linux** (beyond Security)
- Increase **dependability & reliability** for whole Linux ecosystem
- **Various use cases**: Aerospace, Automotive, Medical & Industrial
- Supported by major **industrial grade Linux distributors** known for mission critical operation and various industries representatives
- Close community collaboration with **Xen, Zephyr, SPDX, Yocto & AGL** projects
- **Reproducible system** creation from specification to testing
- SW **elements**, engineering **processes**, development **tools**

ELISA : Architecture | Processes | Features | Tools | Systems

ELISA
Enabling Linux in Safety Applications

Limitations! The OSS projects collaboration ...

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

But...

Projects provide a path forward and peers to collaborate with!

ELISA
Enabling Linux in Safety Applications

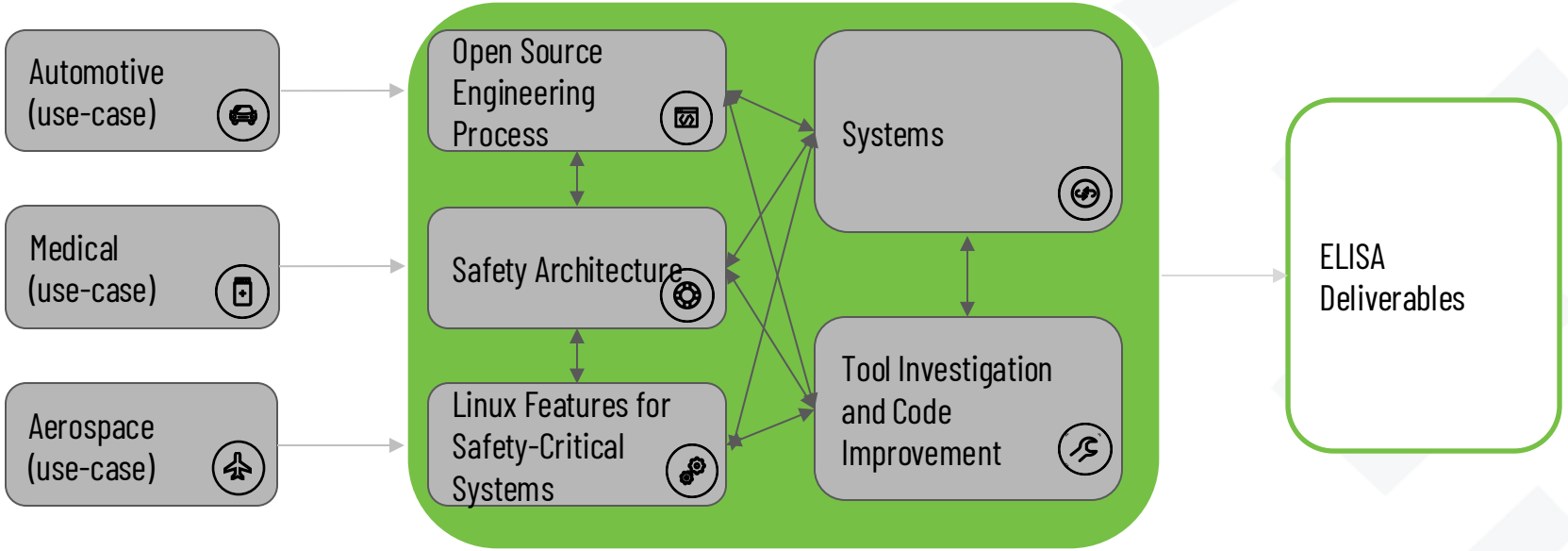
Premier Members

General Members

Associate Members

Industry Support

Different industries share the same core



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.
- Tools + Documentation help to understand complex systems better.

ELISA ease your path with tools and improved documentation

ELISA
Enabling Linux in Safety Applications

Part2: Initiatives and way forward

ELISA Working Groups - Deliverables

- Elements / Software
  - meta-elisa
  - Reproducible system ⭐ 3.
- Processes
  - STPA
- Tools
  - Codechecker
  - Workload tracing ⭐ 1.
  - ks-nav ⭐ 2.
- Documentation
  - RT Linux
  - GitHub / Gdrive / Blog / Whitepaper

ELISA Working Groups - Deliverables

- Elements / Software 
- Processes 
- Tools 
- Documentation 

Workload tracing

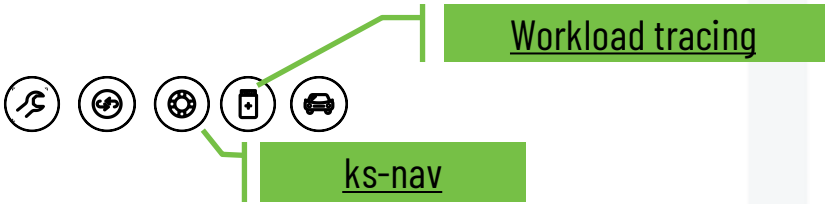
ELISA
Enabling Linux in Safety Applications

Dynamic analysis: Workload tracing documentation mainline.

- Understanding system resource necessary to build and run a workload is important
- Linux tracing and strace can be used to discover the system resource in use by a workload
- Additional tools (like perf, stress-ng, paxtest) can help to analyze performance and security of the OS
- *Credits to Shuah Khan & Shefali Sharma for bringing it mainline*
  - /Documentation/admin-guide/workload-tracing.rst

ELISA Working Groups - Deliverables

- Elements / Software
- Processes
- Tools
- Documentation

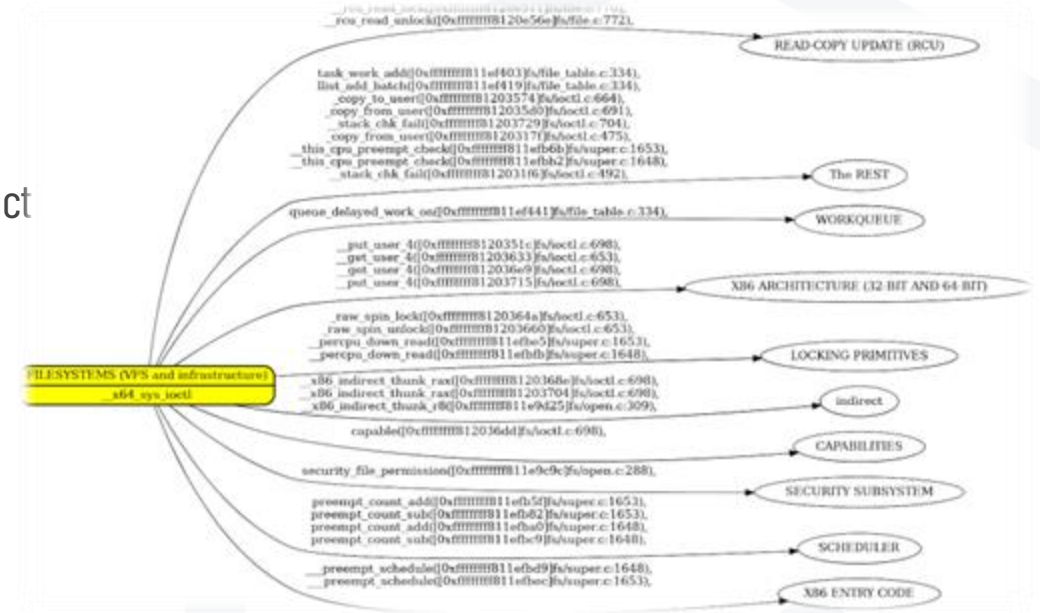


ELISA
Enabling Linux in Safety Applications

Static analysis: e.g. by ks-nav (Kernel Static-Analysis Navigator)

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

- Discussion ongoing to upstream the tool inside Kernel ecosystem
- Credits to Alessandro Carminati & Maurizio Papini (both Red Hat)



https://github.com/elisa-tech/ks-nav

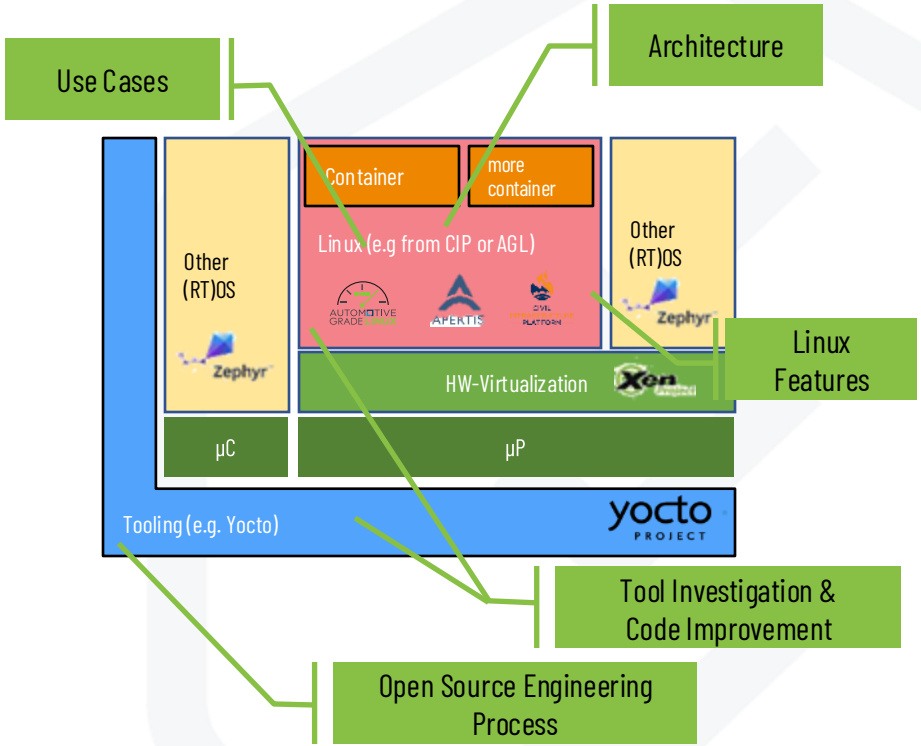
ELISA Working Groups - Deliverables

- Elements / Software 
  - Reproducible system
- Processes 
- Tools 
  - Workload tracing
  - ks-nav
- Documentation 

ELISA Working Groups - Fit in an exemplary reproducible* systemn

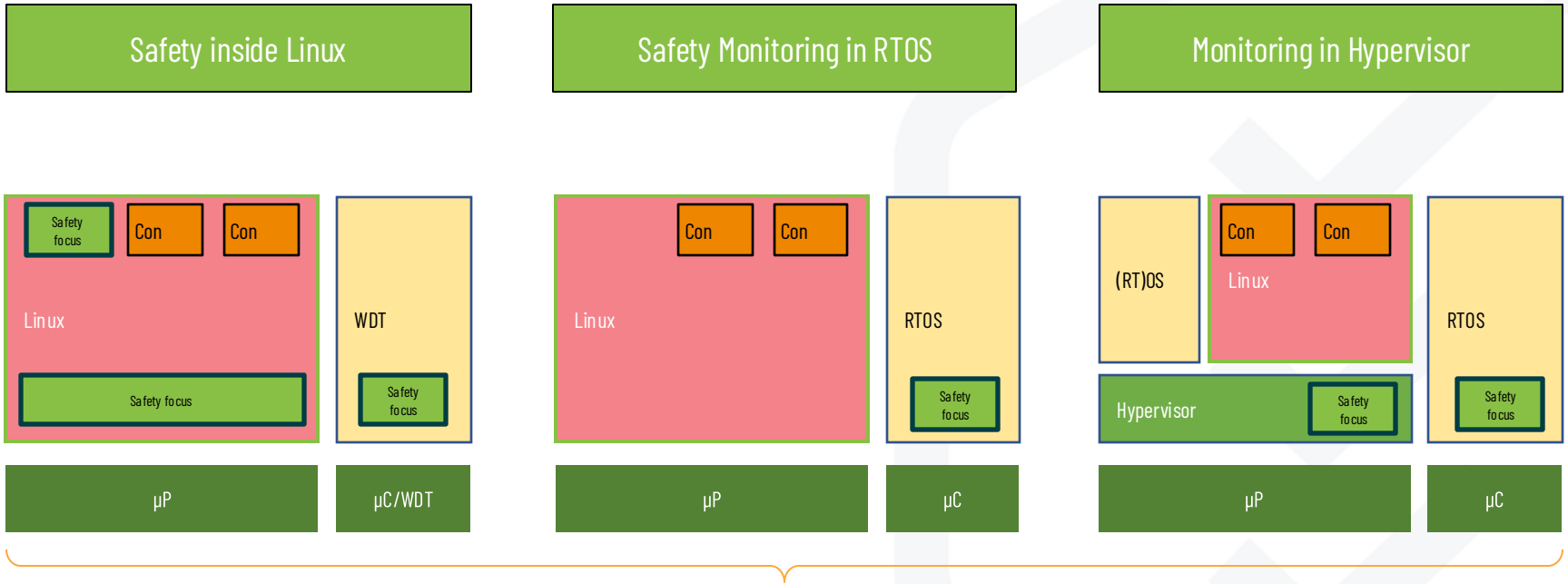
- **Linux Features, Architecture** and **Code Improvements** should be integrated into the reference system directly.
- **Tools** and **Engineering process** should serve the reproducible product creation.
- **Medical, Automotive, Aerospace** and future WG use cases should be able to strip down the reference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

** reproducible: You can recreate the system yourself.*



ELISA
Enabling Linux in Safety Applications

Typical concepts and approaches



Watchdog essential element in various concepts

ELISA
Enabling Linux in Safety Applications

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

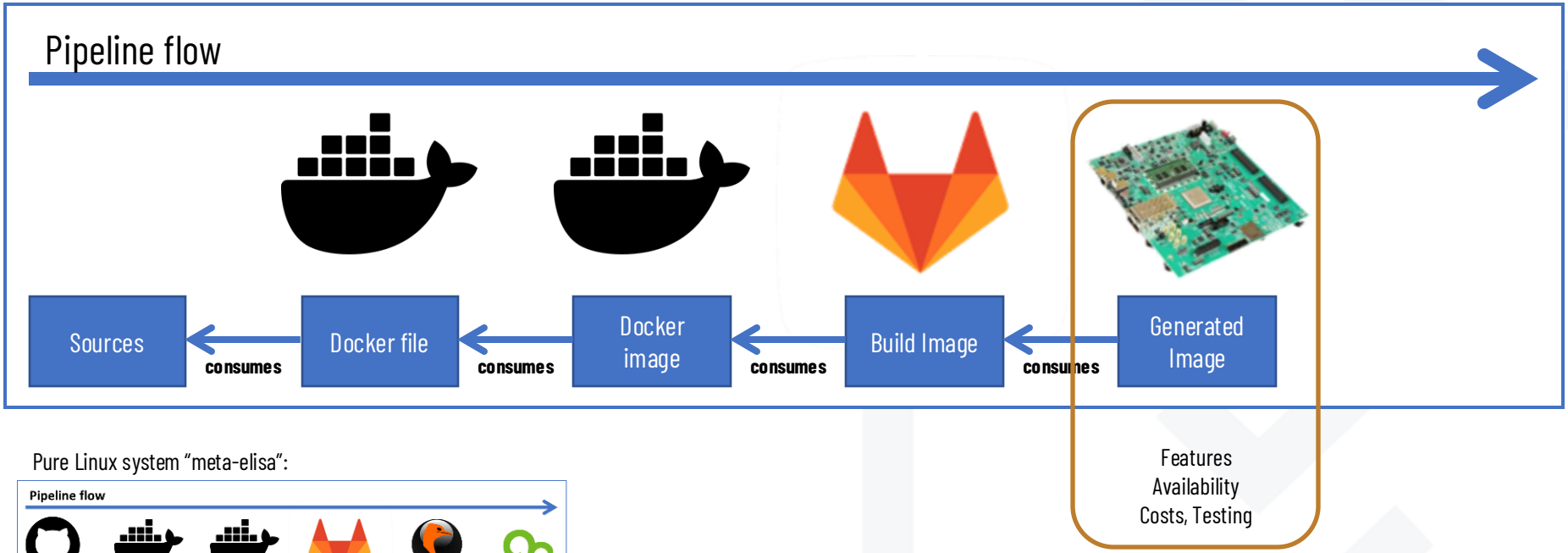
ELISA
Enabling Linux in Safety Applications

Hardware Demonstrator Setup - Xilinx ZCU102

- System elements available from scratch
  - Xen, Zephyr, Linux
  - Good Xen support
- Documentation available in GitHub
  https://github.com/elisa-tech/wg-systems/tree/main/Documentation/xen-demo-zcu102
- Artifacts build in GitLab CI
  https://gitlab.com/elisa-tech/systems-wg-ci (updated daily)
- Qemu derivative exists
- Example system prepared for individual use cases

ELISA
Enabling Linux in Safety Applications

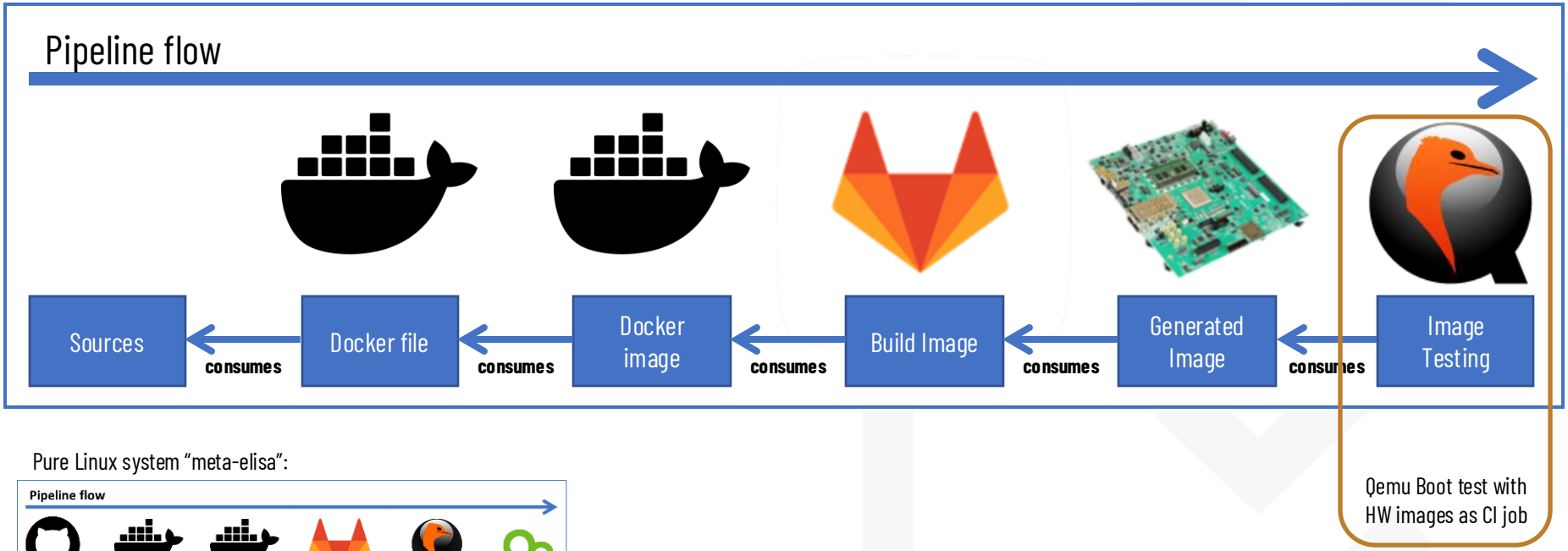
Systems-WG-CI existing build



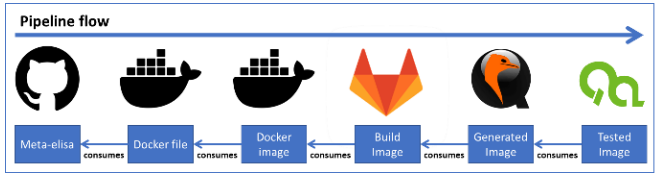
Pure Linux system “meta-elisa”:

ELISA
Enabling Linux in Safety Applications

Systems-WG-CI Enhancements (under development)



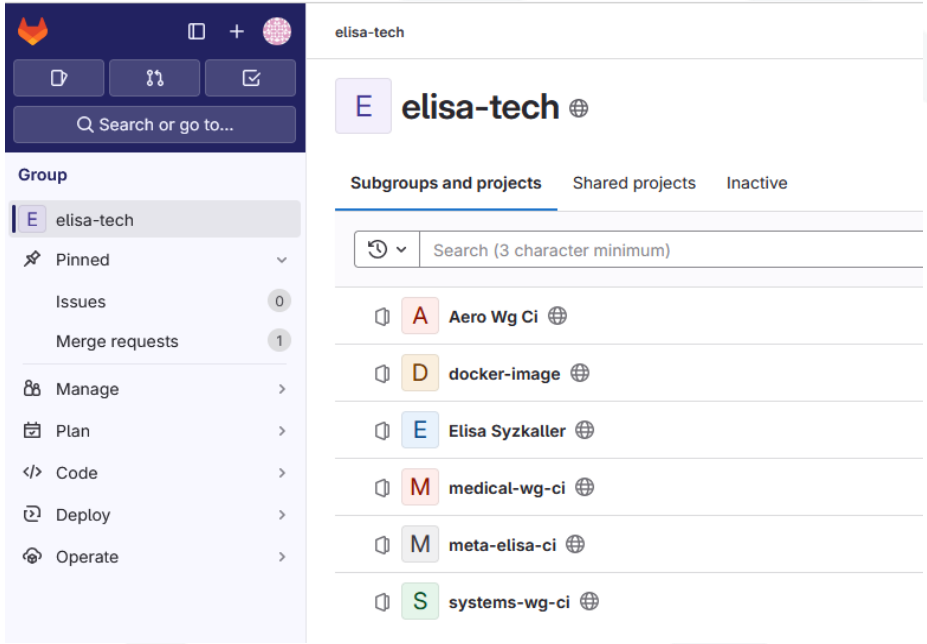
Pure Linux system "meta-elisa":



ELISA
Enabling Linux in Safety Applications

ELISA GitLab CI

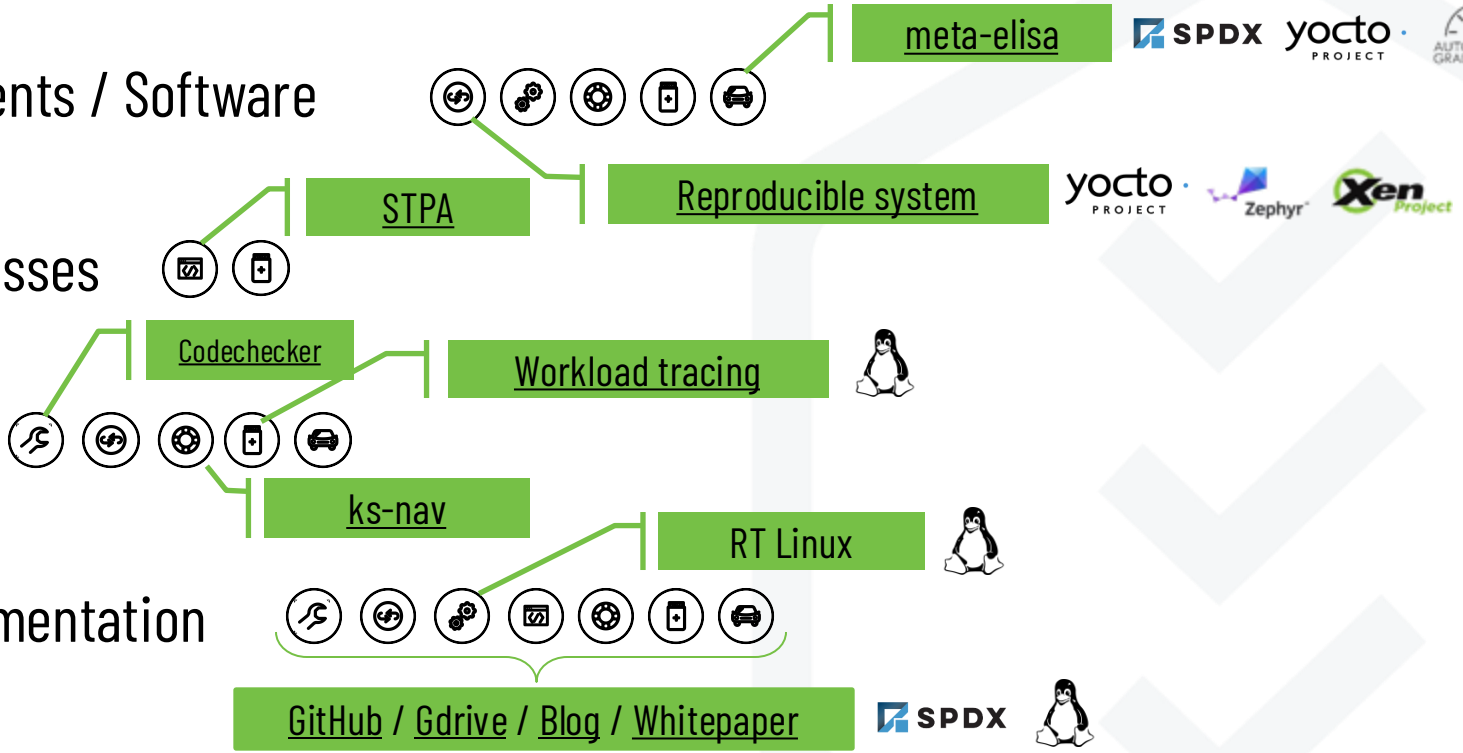
- Different CI builds enabled
  - ZCU102 build for Systems
  - meta-elisa-ci to enhance AGL demo for Automotive WG
  - Boeing minimal kernel example build into Aero WG CI
- Upcoming parts:
  - Minimal kernel configuration on ELISA level
  - SBOM generation for system elements
    (AGL SBOM already enabled)



https://gitlab.com/elisa-tech/

ELISA Working Groups - Deliverables

- Elements / Software
- Processes
- Tools
- Documentation



ELISA
Enabling Linux in Safety Applications

ELISA interactions across the communities

- Open source projects focusing on safety-critical analysis

- Open source projects with safety-critical relevance and comparable system architecture considerations

- Further community interactions

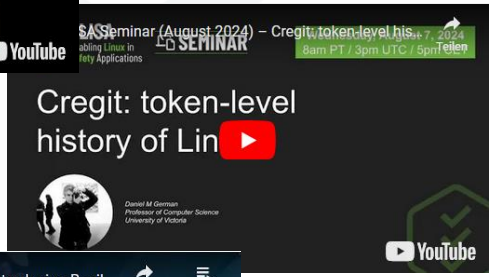
"If you have an apple and I have an apple and we exchange these apples then you and I will still each have one apple. But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas.

— George Bernard Shaw

ELISA Seminar Series

- https://elisa.tech/seminar-series/
- Training & Awareness
- Inside ELISA & outside

- Linux (PREEMPT_RT, page table, ...)
- Safety process (SEooC, Automotive, Avionics, ...)
- Tools (BASIL, cregit, RTLA, ...)
- Communities (Xen, stress-ng, KernelCI, ...)



ELISA
Enabling Linux in Safety Applications

Summary: Documenting what is safe is key!

- Safe systems can only be achieved with sufficient understanding of the system and its parts.
- ELISA provides the tools, the processes and enhance documentation
  - to be used by distributors of the Linux kernel
- Resulting documentation feeds into the safety manual.
  - typically provided by a distributor of the Linux kernel (not ELISA 😉)
- ELISA activities serve various industries **not** only safety-criticality systems.

ELISA
Enabling Linux in Safety Applications

- The discussion continues Friday afternoon at Linux Plumbers!
- **Join the safe Systems with Linux MC** (virtually, if you do not have a ticket yet. 😉)
- Details: https://lpc.events/event/18/sessions/187

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
	Improving kernel design documentation and involving experts *"Hall N2", Austria Center*	*Gabriele Paoloni* 17:50 - 18:10
18:00	**Discussion of Next Steps** *"Hall N2", Austria Center*	*Kate Stewart et al.* 18:10 - 18:30

ELISA
Enabling Linux in Safety Applications

Time for questions & discussion!

Getting involved with ELISA

https://elisa.tech

https://github.com/elisa-tech

https://lists.elisa.tech

https://www.youtube.com/@elisaproject8453

ELISA
Enabling Linux in Safety Applications

Getting involved with Zephyr

https://www.zephyrproject.org

https://www.github.com/zephyrproject-rtos

https://lists.zephyrproject.org

https://chat.zephyrproject.org

ELISA
Enabling Linux in Safety Applications

Getting involved with Xen

https://www.xenproject.org

https://github.com/xen-project

https://xenproject.org/help/mailing-list/

https://xenproject.org/help/matrix/

ELISA
Enabling Linux in Safety Applications


THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
EUROPE