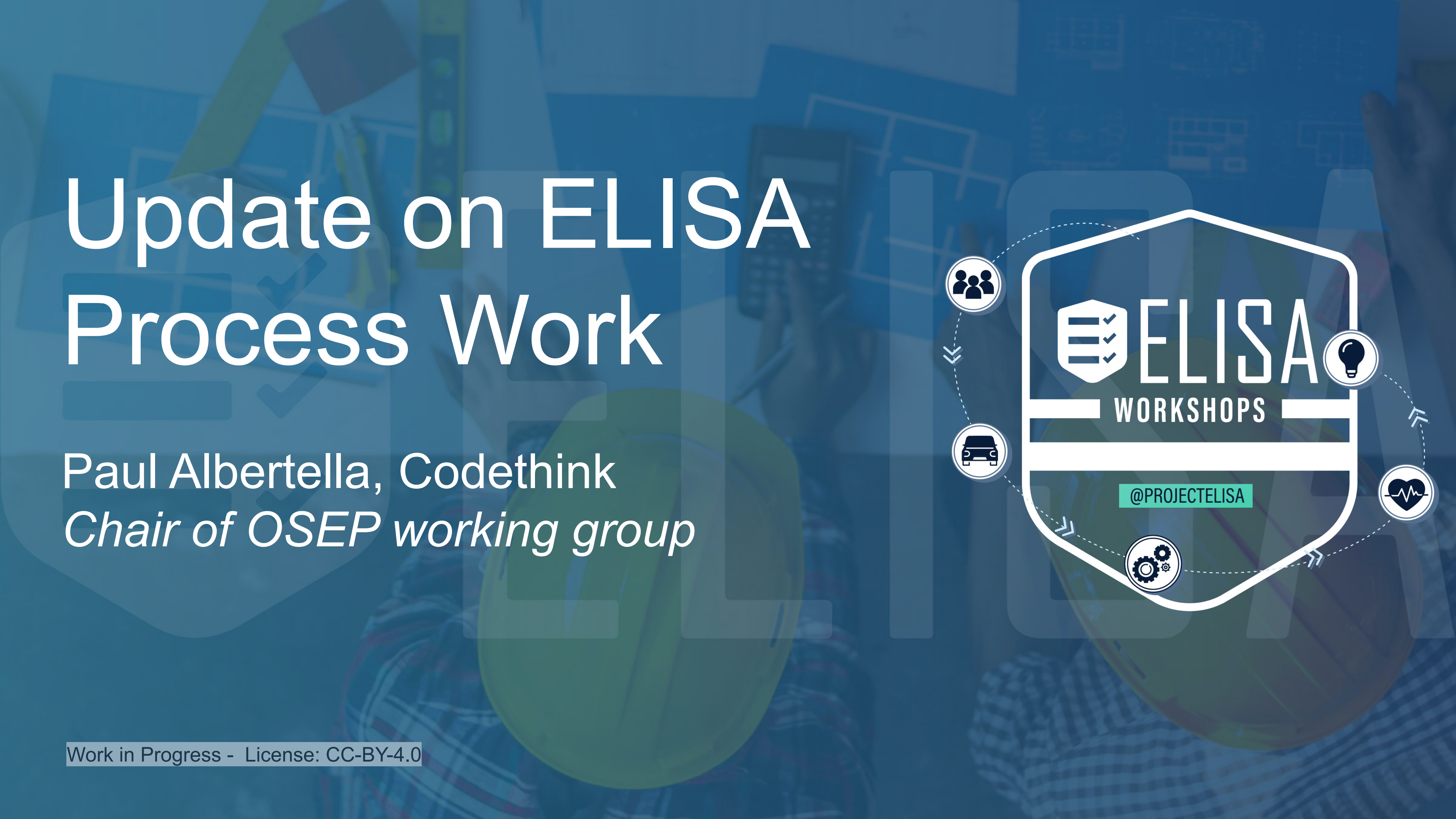
Update on ELISA Process Work

Paul Albertella, Codethink

Chair of OSEP working group

ELISA WORKSHOPS

@PROJECTELISA



Introduction

- OSEP: *Open Source Engineering Process* working group
- Developing common processes and frameworks for ELISA
  - Establish a consistent framing / vocabulary for analysis and discussions
  - Develop safety analysis approaches and system models to enable comparison of results
  - Processes for drafting, reviewing and publishing results
- Have historically attempted to focus on safety analysis
  - What kind of *claims* do we want to make about Linux in the context of safety use cases?
  - How can we describe these safety use cases, and analyse the role that Linux plays in them?
  - Can we use this to derive a common set of safety requirements for Linux?
- Discussions are frequently more wide-ranging!
  - Processes, methodologies, technical topics, basis for safety claims, competency, etc
- Summary of current activities / topics

Why ‘proven in use’ arguments for Linux are invalid

- **Goal**: Document why a ‘proven in use’ argument cannot realistically be applied for the use of Linux in safety-critical systems
  - The fact that Linux is widely trusted in business-critical applications is not a sufficient basis for trusting it in safety-critical applications, or using the ‘proven in use’ argument in standards
  - However, there may be ways that we can use historical data to argue some things e.g. stability of core features, evidence of continuous improvement
  - Drafting a document explaining this - review comments and contributions are welcome: https://github.com/elisa-tech/wg-osep/pull/21
- Questions:
  - What level of detail is required?
    - Is this an introduction or a detailed position statement?
  - Where should the completed document be published?
    - See later *Review and publication of results* slide

Documenting ‘failure modes’ or ‘risk factors’

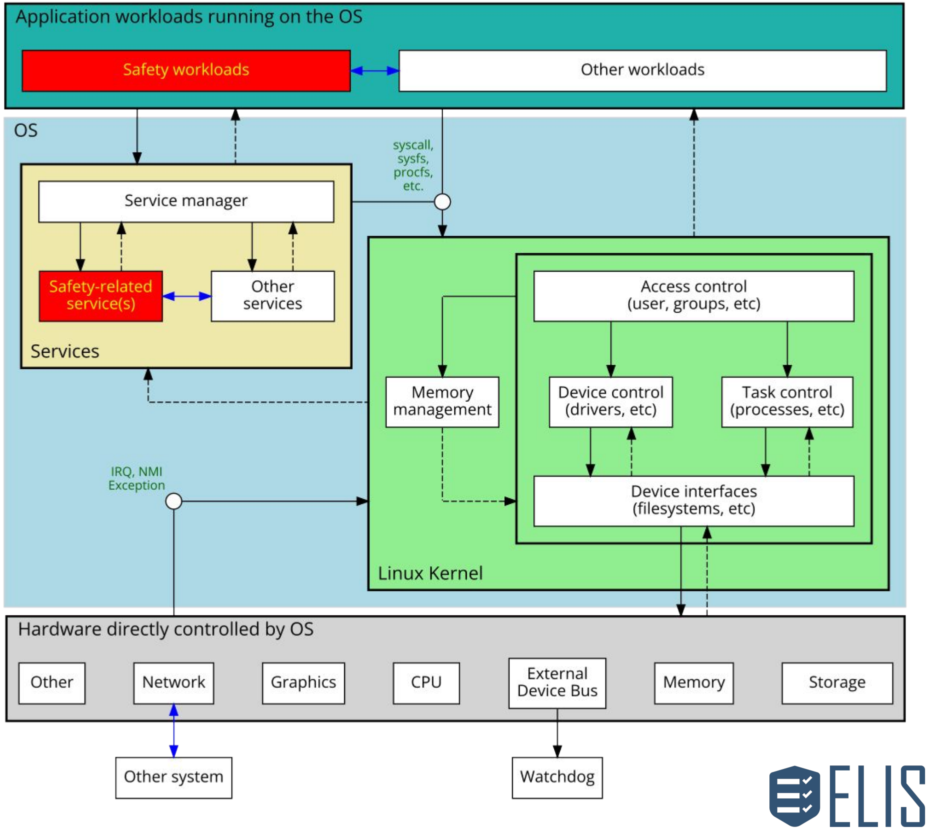
- **Goal**: Document limitations and/or potential weaknesses in the Linux kernel design or implementation, which must be considered as part of any safety analysis
  - As proposed by Igor Stoppa in the previous session
  - Curated list of ‘risk factors’ to consider when configuring the kernel and designing system- and application-level mitigations for a safety-relevant system involving Linux
- **Questions**
  - Can we systematically identify these for the whole kernel, or for a defined configuration?
    - e.g. Using a methodology like FMEA or STPA
  - How should we characterise and document the results?
    - ‘Failure modes’ might be misinterpreted as criticism, implying bugs or poor quality
    - Each factor should be documented in a consistent form, to make the list easier to use and maintain
    - Needs to be meaningful for different audiences (safety analysts, kernel developers, system designers)
  - Do we consider only ‘credible faults’ or consider an active attacker exploiting weaknesses?
    - Bringing security into the equation as well as safety

Modelling safety role(s) of Linux in a system

- **Goal**: Define a model for the role(s) that Linux might have in a safety-related system
- Initial set of categories proposed
  - No role in any safety scenario, other than as a source of interference
  - Active role in a safety function, but no responsibility for ensuring that it is correct
  - Responsibility for some parts of a safety function or functions
  - Responsibility for all safety functions
- **Questions**
  - How do we intend to use these role categories?
  - Need to consider *availability* as well as *correctness* when identifying roles and responsibilities with respect to safety functions
  - Can we frame in terms of *levels of trust* placed in Linux in a given system context?
  - Do we consider only ‘credible’ faults or consider an attacker actively exploiting weaknesses?
    - Brings cybersecurity into the equation as well

Modelling behaviour of Linux as part of an OS

- **Goal**: Define an abstract model for OS and kernel functions in a Linux-based operating system, to provide a consistent framework for analysis and documentation of risk factors
- PR for review / discussion: https://github.com/elisa-tech/wg-osep/pull/19
- **Questions**
  - Is this model sufficient as a framework for analysis?
  - Combine with previous model to show where safety responsibility is assigned in a system?


Application workloads running on the OS
Safety workloads
Other workloads
OS
syscall, sysfs, procfs, etc.
Service manager
Safety-related service(s)
Other services
Services
Access control (user, groups, etc)
Memory management
Device control (drivers, etc)
Task control (processes, etc)
IRQ, NMI Exception
Device interfaces (filesystems, etc)
Linux Kernel
Hardware directly controlled by OS
Other
Network
Graphics
CPU
External Device Bus
Memory
Storage
Other system
Watchdog

Review and publication of results

- **Goal**: Establish common processes for creating, reviewing and publishing documents and diagrams describing results of ELISA workgroup discussions and analysis
  - Contribution guidelines, including use of GitHub
  - Review and approval process, including criteria to be used by reviewers
  - Publication process and format once accepted into mainline
- Questions:
  - Where / how to publish?
    - As blog posts (perhaps linking to repo)?
    - As web content rendered from repos via GitHub Pages?
  - Competency and affiliations of authors and reviewers
    - ‘Mini-CVs’ for contributors, describing employers, experience, areas of expertise, etc?
    - Standard legal and commercial disclaimers

Discussion and next steps

Licensing of Workshop Results

All work created during the workshop is licensed under *Creative Commons Attribution 4.0 International (CC-BY-4.0)* [https://creativecommons.org/licenses/by/4.0/] by default, or under another suitable open-source license, e.g., GPL-2.0 for kernel code contributions.

You are free to:

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.