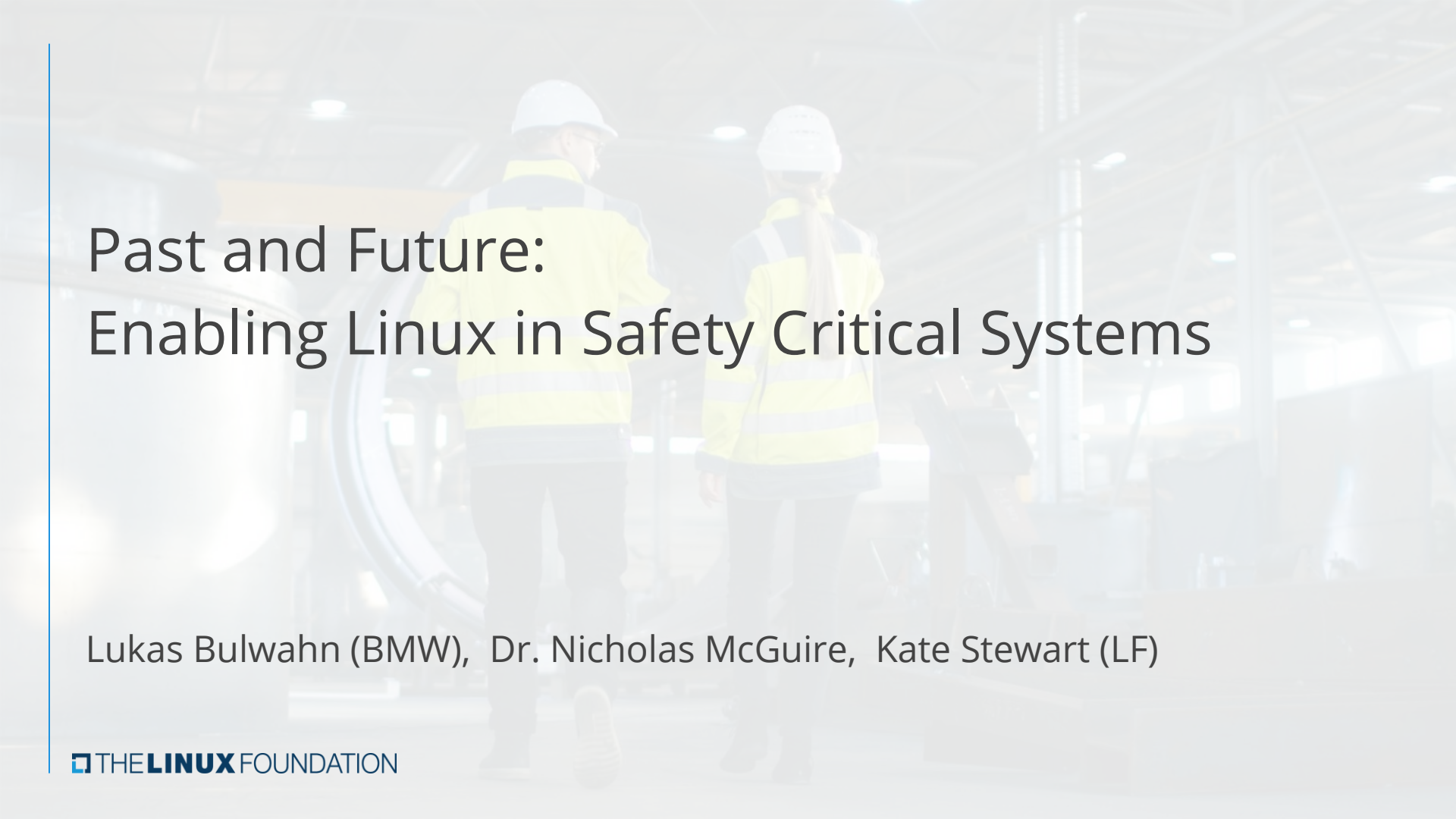


Past and Future:
Enabling Linux in Safety Critical Systems

Lukas Bulwahn (BMW), Dr. Nicholas McGuire, Kate Stewart (LF)

Motivation to use Linux in Safety-Critical Systems

Business Motivation

THE WALL STREET JOURNAL.

ESSAY

Why Software Is Eating The World

By Marc Andreessen

August 20, 2011

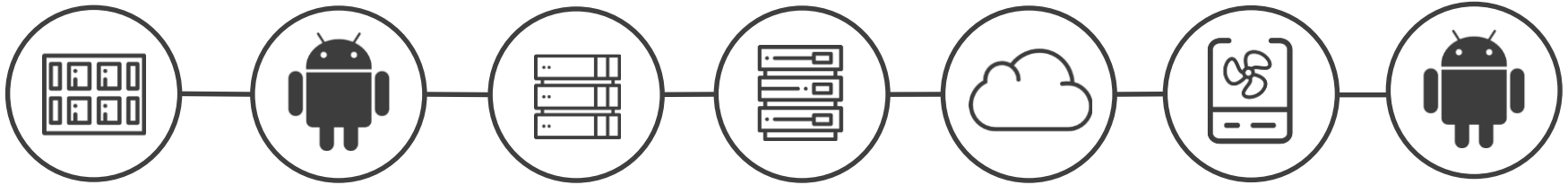
Main message: Software innovations will disrupt every industry, even very established industries.

Companies in the mechatronics industry are struggling with this change which is being driven by software innovations and software industry competitors.

Give profits to the software vendors or invest to explore alternatives?

The Linux Operating System Has Grown into the Most Important Software Platform in the World

100%	82%	2nd	90%	90%	62%	#1
Supercomputer Market	Smartphone Market Share	To Windows in Enterprise	Mainframe Customers	Public Cloud Workload	Embedded Systems Market	Internet Client



Every market Linux has entered it eventually dominates

Assessing Linux for Use 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

The Linux kernel is missing:

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

Can these gaps be closed?

OSADL SIL2LinuxMP Project

Mission:

- Provide procedures and methods to qualify Linux on a multi-core embedded platform at safety integrity level 2 (SIL2) according to IEC 61508 Ed 2.
- Show feasibility of procedures and methods on a real-world system
- Show potential for collaboration and re-use of Linux kernel analysis

Project running since 2015, organized by OSADL

OSADL SIL2LinuxMP Project

Collaborative project of industrial & research partners

- **Full members:** BMW Car IT, Intel (since ‘17), A&R Tech, KUKA, Sensor-Technik Wiedemann (full members till ’16, reviewing members in ‘17)
- **Reviewing members:** Bosch, Elektrobit, Hitachi, Linutronix, MBDA Italia, MEN Mikro Elektronik, Mentor, OpenSynergy, Pilz GmbH & Co. KG, Renesas, Vienna Water Monitoring Solutions
- **Academic members:** A. Khoroshilov (ISP RAS), K. Chow (Lanzhou Univ.), J. Lawall (Inria/LIP6),
  F. Tränkle (HS Heilbronn)
- **Experts from certification bodies:** B. Nölte (TÜV Süd), O. Busa, R. Heinen, H. Schäbe (TÜV Rheinland)
- **SIL2LinuxMP core working team:** N. McGuire, A. Platschek, L. Böhm, M. Kreidl (OpenTech)

Introduction to Functional Safety

Functional Safety

"**Functional safety** is the part of the overall safety of a system (...) that depends on the system (...) **operating correctly in response to its inputs**, including the safe management of likely **operator errors, hardware failures and environmental changes**.

The **objective of functional safety** is freedom from **unacceptable risk** of physical injury or of damage to the health of people either directly or indirectly."

(Source: wikipedia.org:Functional Safety)

Work on Functional Safety is **Risk Management**

- ➔ Risk Management is to **focus quality assurance on the right aspects and right parts!**
- ➔ It is NOT to do just more work or write hundreds of documents!

Functional Safety

How to determine the **acceptable risk?**

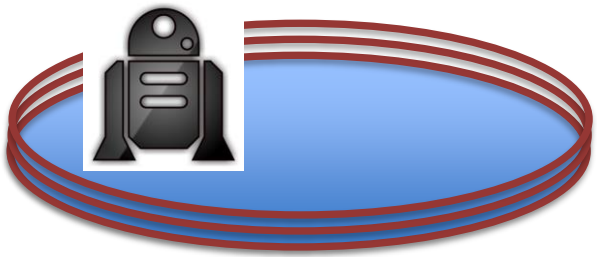
- ➔ Agreement in **global safety standards**
- ➔ **IEC 61508**: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems - a basic functional safety standard applicable to all kinds of industry (Adaptations include: ISO 26262 for automotive, IEC 62279 for railway applications)

How to **design safe systems?**

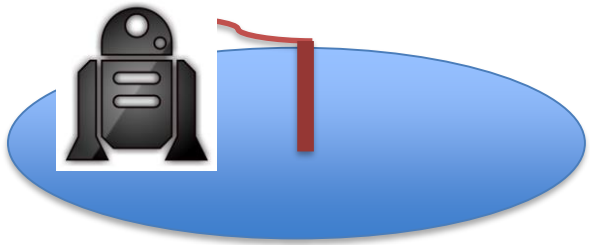
- ➔ **System design & system analysis**
  - ◆ Analyze your system to know which parts must be of high quality for the system's safety
  - ◆ Assign safety integrity levels (SIL) to those parts,
    SIL1 (low safety level) to SIL4 (high safety level)
- ➔ **Rigorous development process**
  - ◆ Develop those parts with high SIL with sufficient rigor (= the right development process)
  - ◆ Safety standards state which objectives shall be achieved in each development phase

Functional Safety by Example - part 1

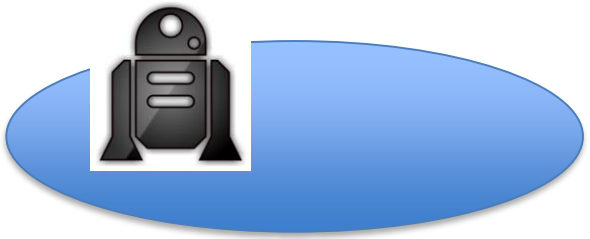
The robot can only harm somebody when it leaves the blue area.



Is this system safe?



Is this system safe?

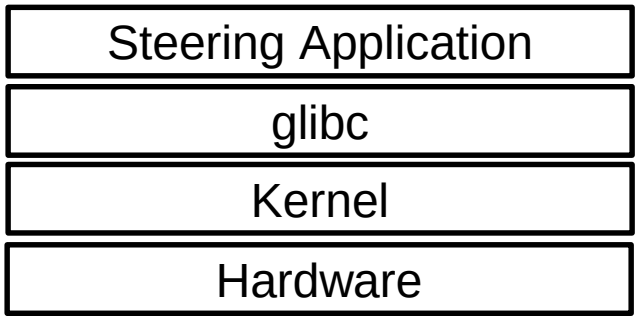


Is this system safe?

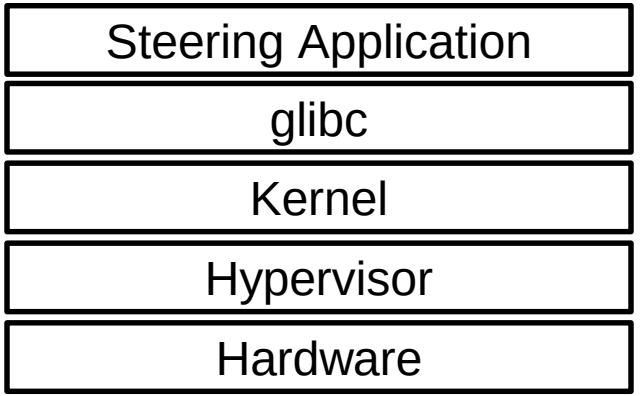
To assess whether your system is safe,
you need to understand your system sufficiently.

Functional Safety by Example - part 2

The robot can only harm somebody when the steering application steers the robot to leave the blue area.



Is this system safe?



Is this system safe?

To assess whether your system is safe,
you need to understand your system sufficiently.

Linux in Safety-Critical Applications

Safety-Critical Linux

To assess whether your system is safe,
you need to understand your system sufficiently.

➔ If your system‘s safety depends on Linux, you need to understand Linux sufficiently for your system context and use

Safety by Process Argument

Compliance to Objectives of Safety Standards by Development Process Assessment:

- ➔ Linux has been continuously developed for 27+ years
- ➔ Continuous Process Improvement is in place.
  - ◆ When technical or procedural issues in the kernel development are identified and pressing, the community addresses them.
- ➔ Evidence for the requisite process quality and process improvement quality exists already.
- ➔ This evidence can indicate that all objectives of a safety integrity level 2 for selected parts and properties are met.

Safety-Critical Linux Process Approach

The difference between Safety-Critical Linux and main-line Linux **is the way you use it**.

- *Understand* your system and *understand* Linux
- Make sure your system uses Linux based on the *selected* properties of Linux where *you can assure* quality exists already.

How to Make Linux-Based Systems Safe

Organisation‘s shared knowledge of the system and Linux makes the system safe

- ➔ *Processes and Methods* to understand:
  - the qualities of the complex software system
  - the qualities of the Linux kernel
- ➔ Education on these topics is the key to your safety product development.

SIL2LinuxMP Project Retrospective

Effective exchange and education of ideas & challenges:

- ➔ A defined plan and compliance route
  - ◆ Reviewed by project participants and a safety authority

- ➔ Some first technical investigations:
  - ◆ System engineering methods for complex software systems
  - ◆ Methods and tools for kernel investigations and gathering process evidence
  - ◆ Understanding existing Linux kernel verification tools

Things to Keep

What was important and what went well?

- Education and exchange of ideas in eight three-day workshops
  - System safety engineering for complex software system
  - Interpretation of the IEC 61508 for pre-existing software
  - Relevant verification tools applied in the Linux kernel development

Lessons Learned

What we need to address in any future collaboration?

- Organised as research project, not as collaboration
- Underestimated collaboration around functional safety
  - Difficult & mind-bending field, different from software engineering
  - Open Collaboration in functional safety was not established
- Misunderstanding of educational goal
- No suitable hardware and access to documentation suitable for collaboration
- Members with little participation had difficulties to make use of results

From Research to Collaboration

Collaboration Goals

Shared development and effort on:

- Understanding safety engineering of complex systems
- Creating risk assessments of the kernel subsystems and features
- Gathering evidences of kernel development process compliance
- Developing supporting tools
- Creating material to train and educate engineers

Key Elements of an Effective Collaboration

- ➔ Establish well-defined governance and project steering in a neutral organisation
- ➔ Maintain good community health
- ➔ Keep educating on functional safety and process assessment
- ➔ Share a common system to focus on common activities
- ➔ Reach out to Linux and safety communities, as well as to hardware vendors

The Linux Foundation has a positive track record to establish such collaboration models in various industries.

Successful Outcome of an Effective Collaboration

Assets for safety certification of Linux-based systems

- consisting of a **complete process**, selected kernel features and tools, and previous process assessments
- **shown feasible** with a reference system(s)
- **usable** by properly educated system integrators
- **maintained** over industrial-grade product lifetimes
- **well-known and accepted** by safety community, certification authorities and standardization bodies in multiple industries
- **positively recognised** and impacting the Linux kernel community
- **hardware collateral** from multiple supporting vendors

Limits to the Collaboration

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 it will provide a path forward and peers to collaborate with!

Conclusion

Summary

- We know many of the questions and some of the conceptual answers.
- A number of potential solutions have been prototyped.
- Assessing -stable (or -cip) kernel for GNU/Linux based safety related system seems doable - but not trivial
- This initiative will NOT deliver a pre-certified Linux
  - it will allow reduction of the business risk by building on a understood and accepted compliance approach.
- Some of the tools and methods seem usable for general OSS development

Path forward for “Closing the Gap” is needed

Industry needs an operating system for complex algorithms and software suitable for safety-critical systems

Basis available:

- Functional safety is about managing risk in product development
- Risk in Linux-based systems can only be understood with knowledge of the system and kernel
- Starting point for understanding Linux in safety-critical systems is available

Collaboration proposal:

- Further development of the basis requires industry collaboration
- Technical and organisational proposal is in place

The future of Enabling Linux In Safety Applications is up to all of us...

Thank you for your interest!

Questions?

Next Steps? Learn more and connect with us at safety.linuxfoundation.org