GNU/Linux for safety-related systems - SIL2LinuxMP

Nicholas Mc Guire <safety@osadl.org>

January 28, 2016

Outline

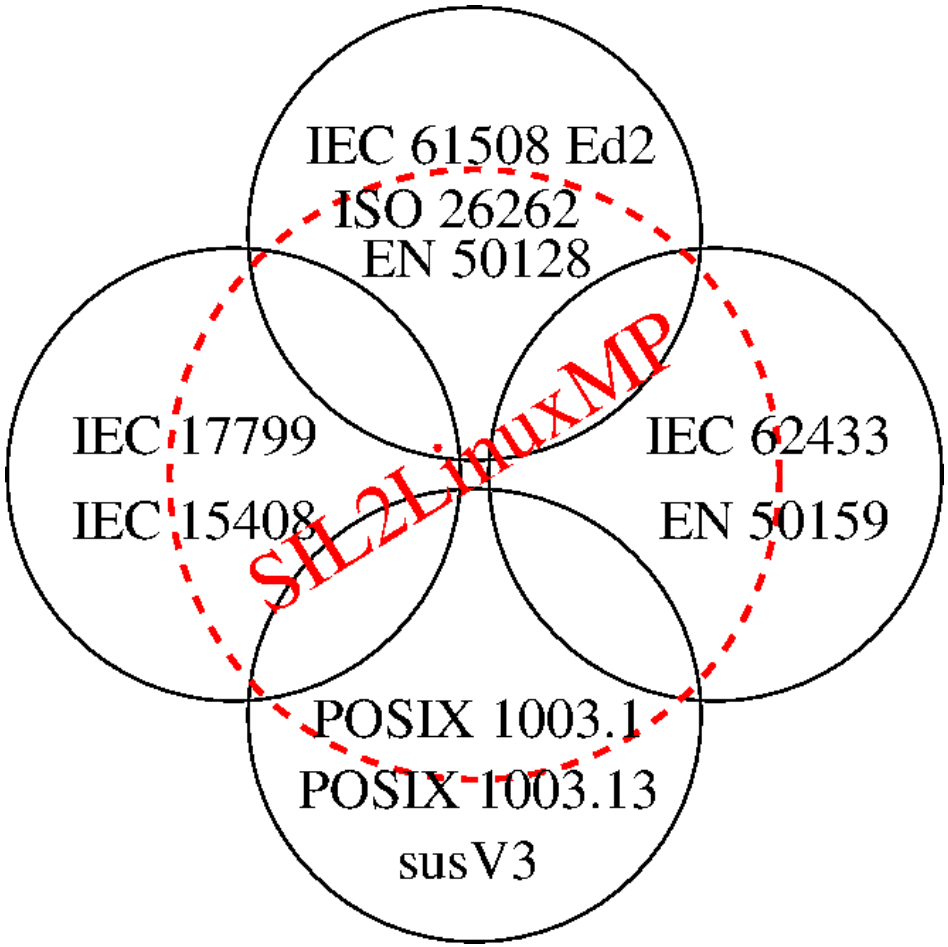
- Context
- Process
- Conclusions

Goal of SIL2LinuxMP

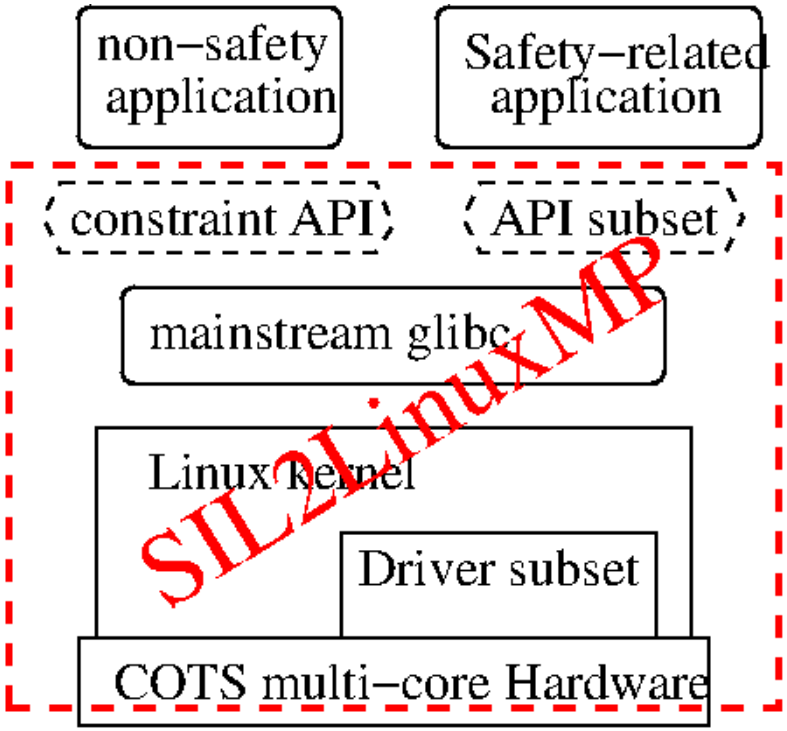
OSADL

- Generic qualification approach
- Suitable for up to SIL2 (IEC 61508 Ed 2)
- Support multicore systems
- Mainline kernel + glibc + tools
- Methods suitable for pre-existing SW intensive systems

SIL2LinuxMP Context



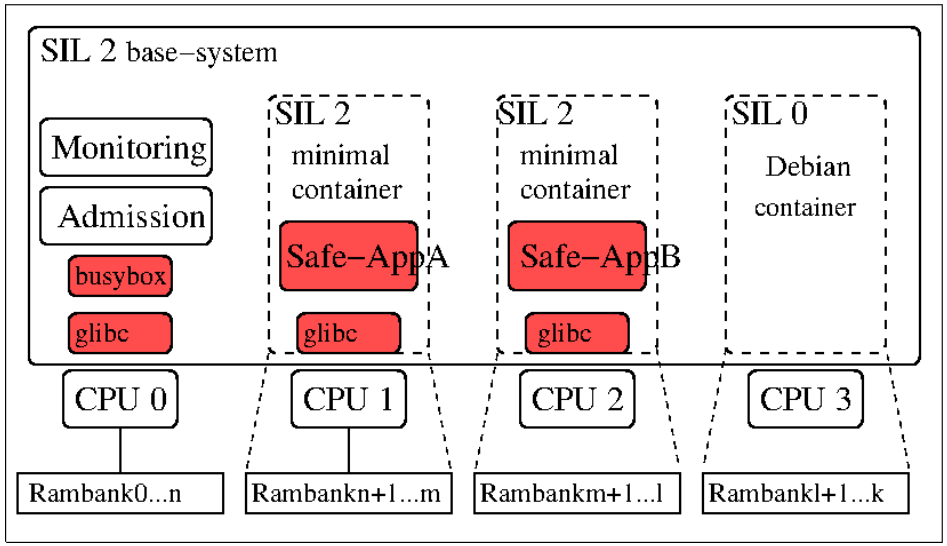
The Goal


non−safety application
Safety−related application
constraint API
API subset
mainstream glibc
Linux kernel
Driver subset
COTS multi−core Hardware
SIL2LinuxMP

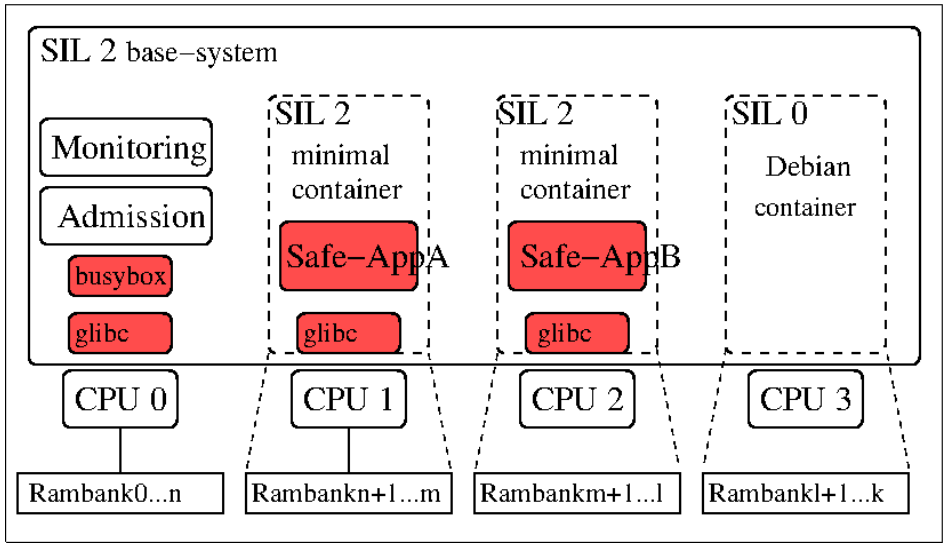
Basic constraints

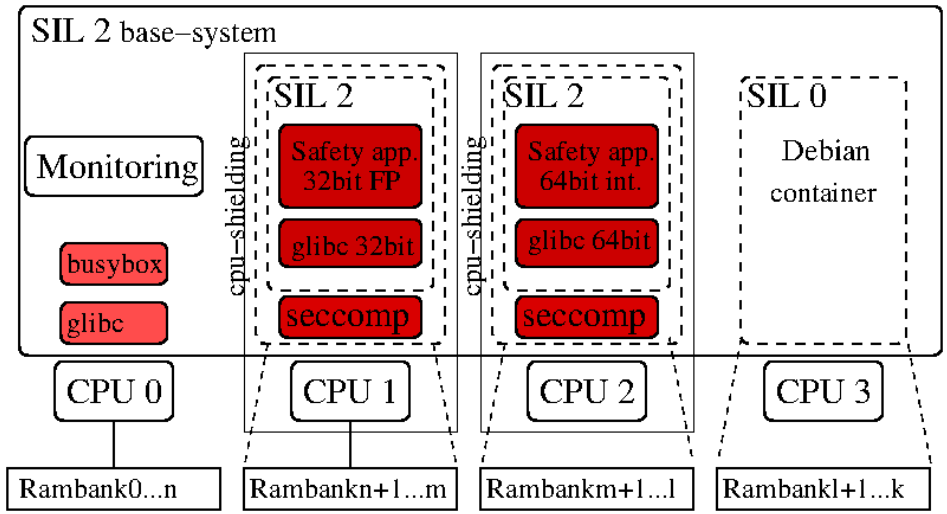
- Minimize kernel <-> follow mainline
- Minimize safety related runtime env
  - glibc
  - busybox runtime environment
  - Handle cgroups "manually" -> minimal launcher
- Compliant development of safety related applications
- Push the full-featured (non-safe) OS into a container
- Minimize/control sharing of resources between safe/non-safe tasks

Arch 4 - prototype architecture



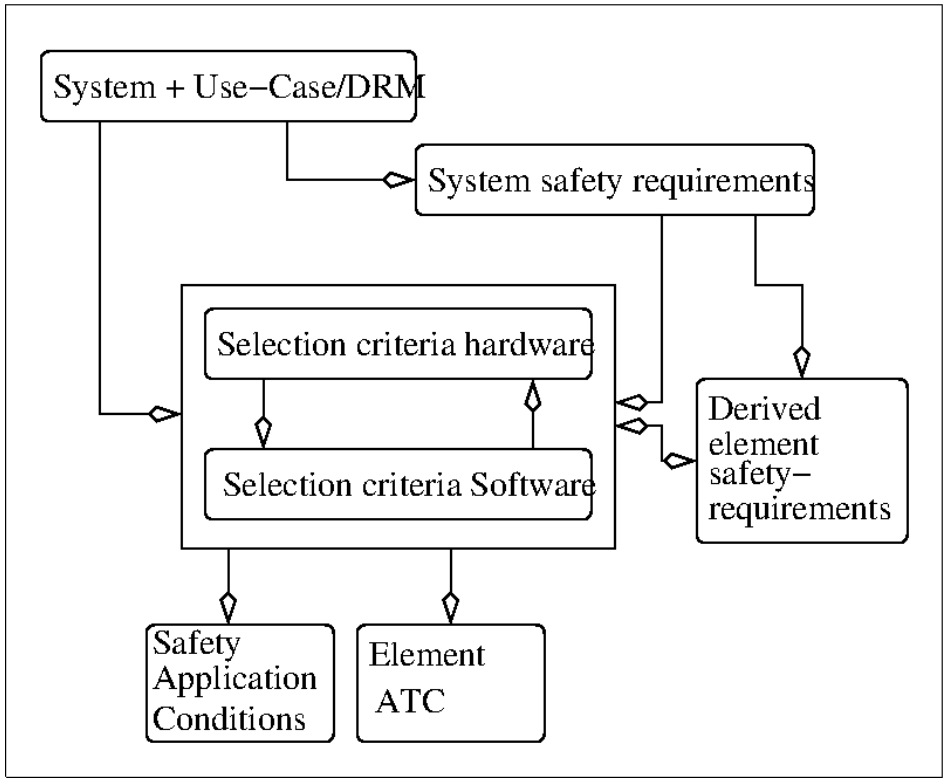
Arch 4 - prototype architecture





think of it as a "distributed system on one chip"

SIL2LinuxMP Selection



Selection has been formalized in the context of 61508-1 Ed 2 as Clause 7.X "E/E/PE safety-related software element selection" - pennding review by TueV Rheinland.

Adjusted software DLC

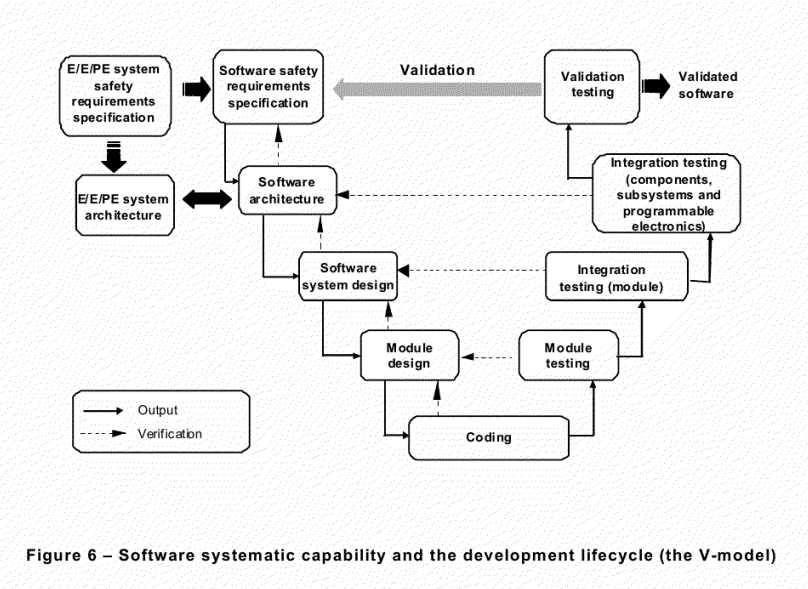


Figure 6 – Software systematic capability and the development lifecycle (the V-model)

Nicholas Mc Guire <safety@osadl.

Adjusted software DLC

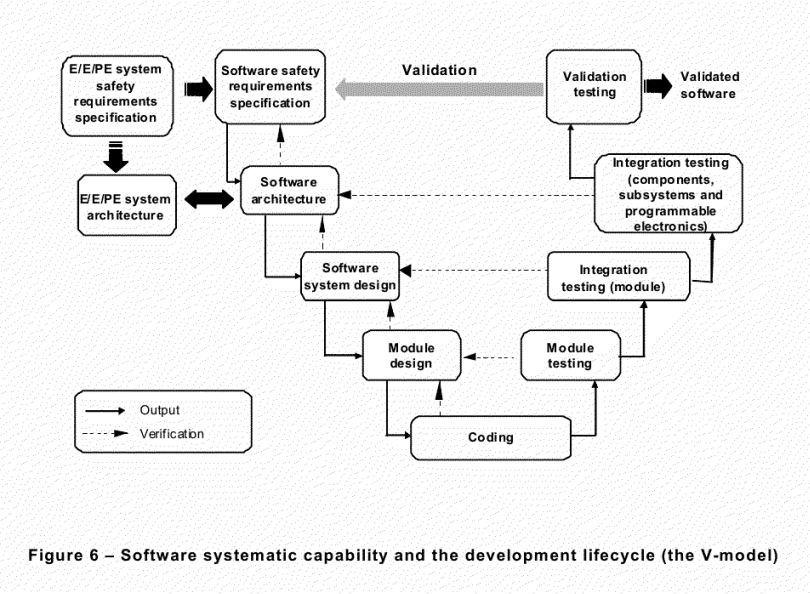
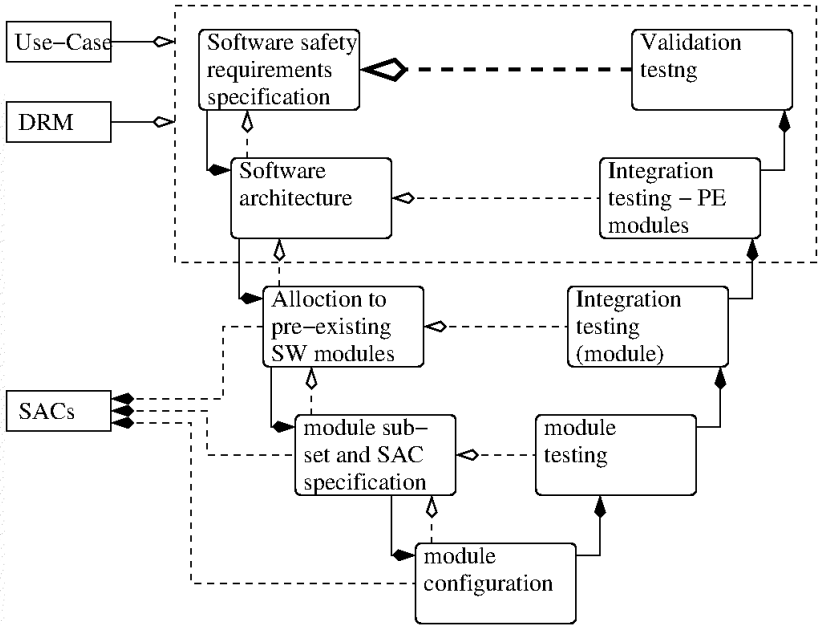


Figure 6 – Software systematic capability and the development lifecycle (the V-model)



Softwar systematic capability – V–model for pre–existing softwa

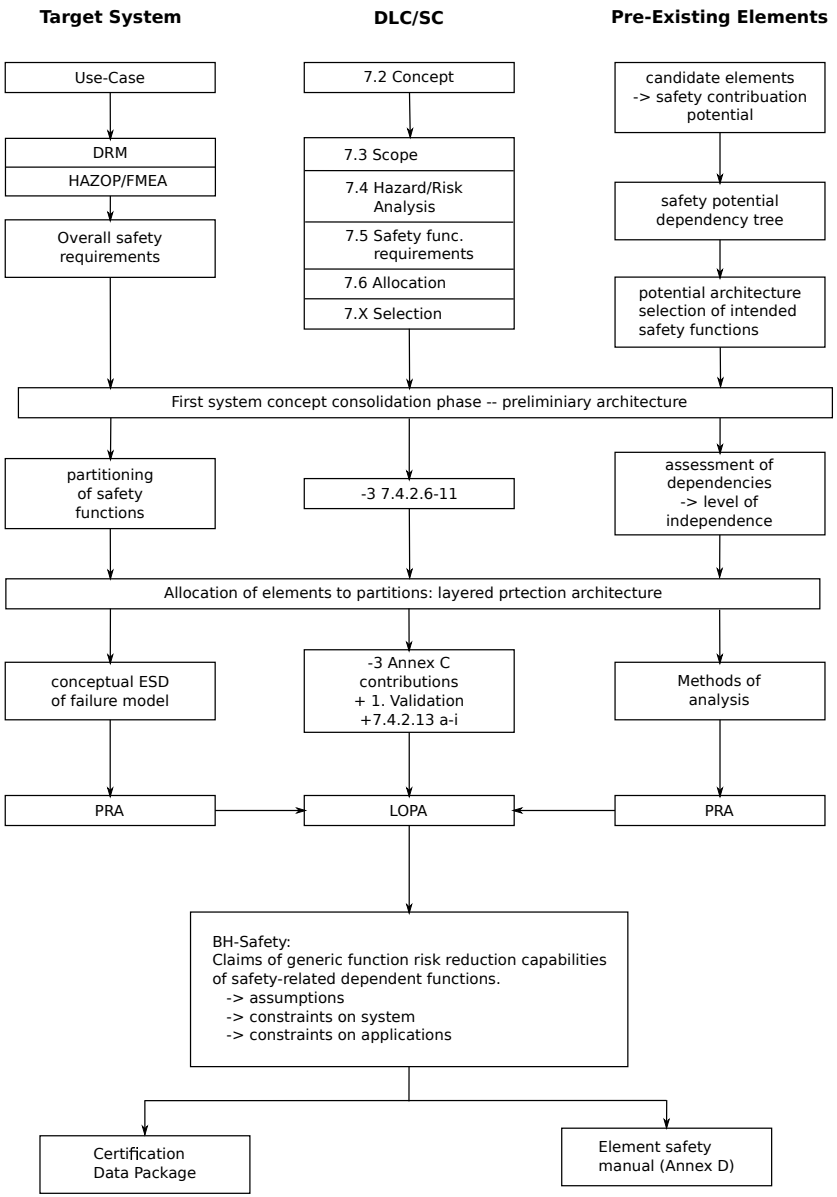
Example: Isolation Techniques

Available technologies to improve non-interference

- Control Groups
- Namespaces
- Seperate filesystem (images/media)
- Replicated glibc/busybox
- Limit system calls (seccomp)
- Real devices managed by core-system
- PALLOC - partitioning allocator
- ABI diversity

Functionality + level of assurance -> safety functional capability

Big picture of DLC/SLC



GNU/Linux for safety-related systems - SIL2LinuxMP

Nicholas Mc Guire <safety@osadl.

Conclusions

- If you want to utilize FLOSS -> fix the processes first
- IEC 61508 was not really conceived with selection as primary strategy in mind - but it **is** doable.
- IEC 61508 is robust enough to provide a solid foundation for formalizing element selection (Route 3_S) as primary strategy
- The process adjustments are in review (TueV Rheinland) ... lets see
- Based on the final processes the method set will be selected
- Applying this to GNU/Linux RTOS will not be trivial - but looks doable

Thanks !

http://www.osadl.org/SIL2