OpenTech

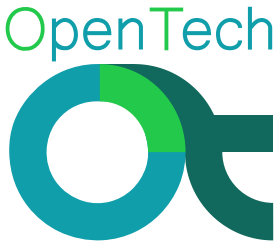
Diversity in SIL2LinuxMP

Diversity approaches investigated for the SIL2LinuxMP architecture

OSADL Safety Critical Linux Working Group

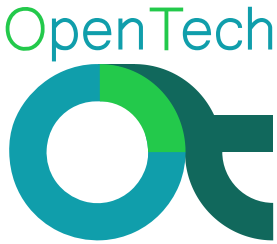
Nicholas Mc Guire
<*safety@osadl.org*>
June 11, 2017

Outline



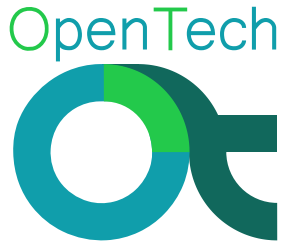
- Context
- Problem Statement
- Solution Space
- Logical Isolation
- Conclusions

61508-3 Ed 2 7.4.2.13
Route 3_S



- Assessment of non-compliant development
- (almost) all of 61508-3 Ed 2 from a different perspective
- Some inconsistencies one needs to work around
- Annex-C plays a key part in guiding interpretation

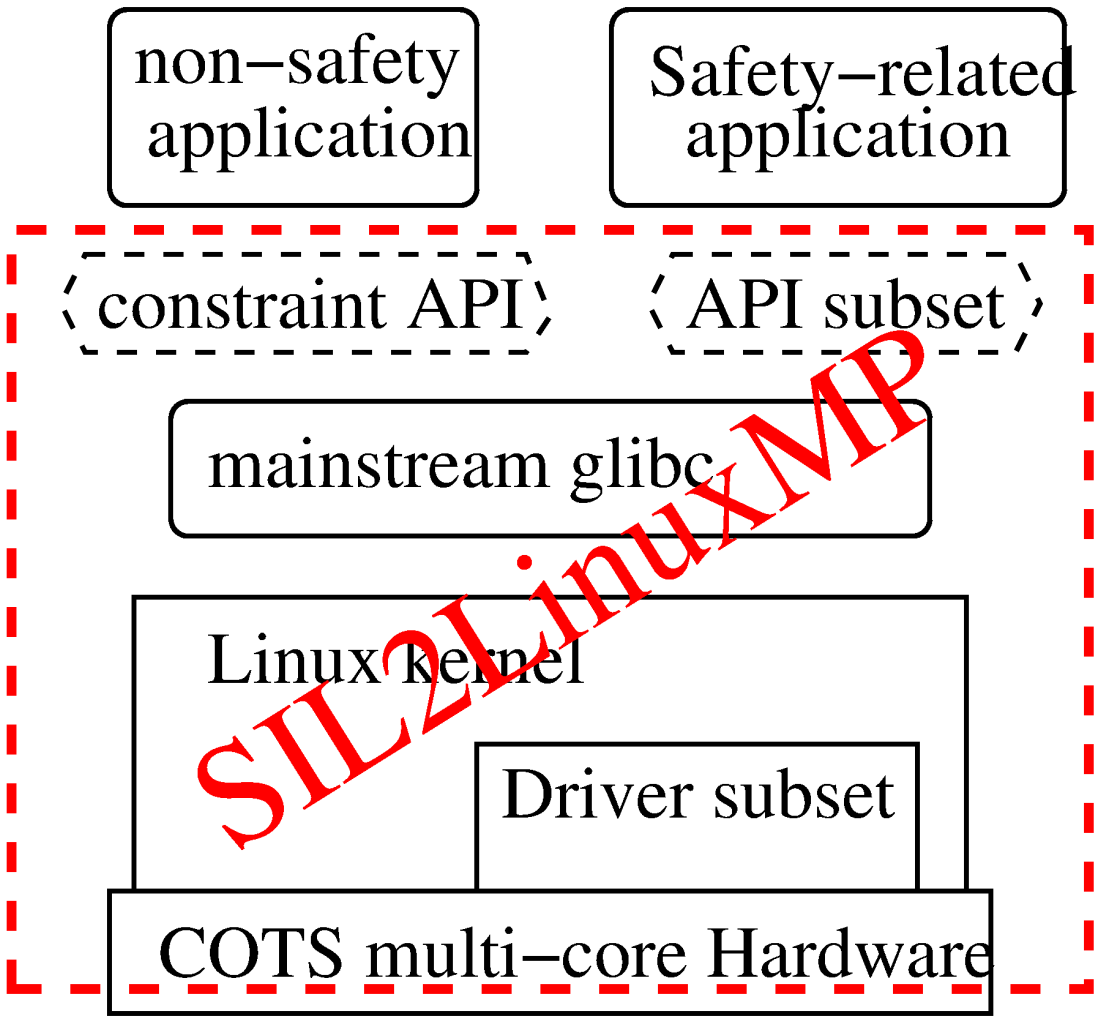
61508-3 Ed 2 7.4.2.13
Route 3_S



- Assessment of non-compliant development
- (almost) all of 61508-3 Ed 2 from a different perspective
- Some inconsistencies one needs to work around
- Annex-C plays a key part in guiding interpretation

has been conceived with a rapidly developing technology in mind; the framework is sufficiently robust and comprehensive to cater for future developments; [61508-3 Ed 2 INTRODUCTION]

Main Elements Overview

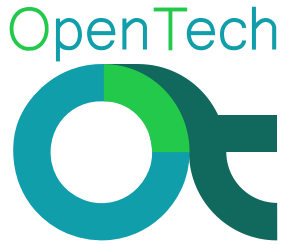


Traditional Metrics Failing

OpenTech

- MC/DC on the Linux kernel ?
- Limited use-of-pointers ?
- Limited use-of-interrupts ?
- Branch coverage - what does it say ?

Traditional Metrics Failing



- MC/DC on the Linux kernel ?
- Limited use-of-pointers ?
- Limited use-of-interrupts ?
- Branch coverage - what does it say ?

Compliance is not about adherence to annexes or tables - it is about achieving the objectives - if the objectives are out of date - the resulting system are not going to be safe.

Pre-existing SW in System context

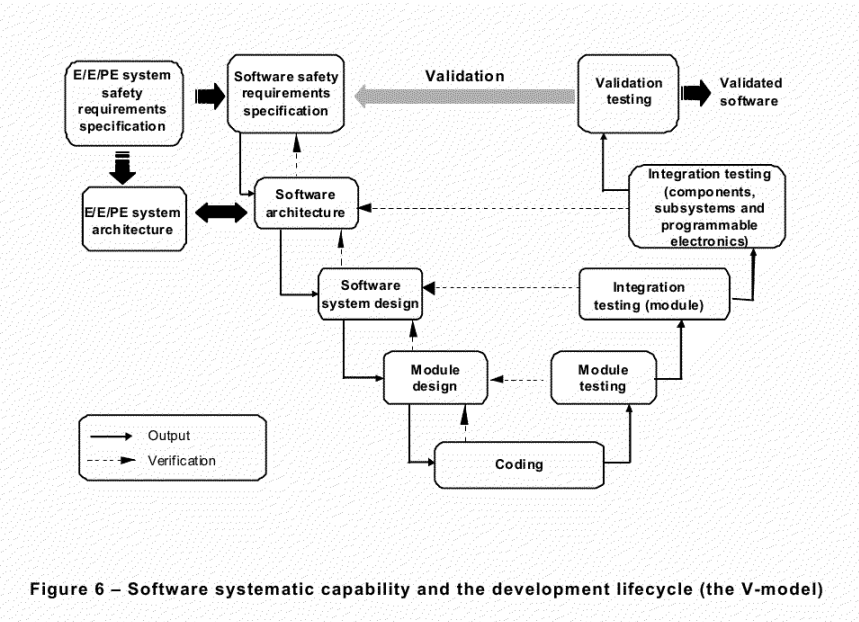


Figure 6 – Software systematic capability and the development lifecycle (the V-model)

Pre-existing SW in System context

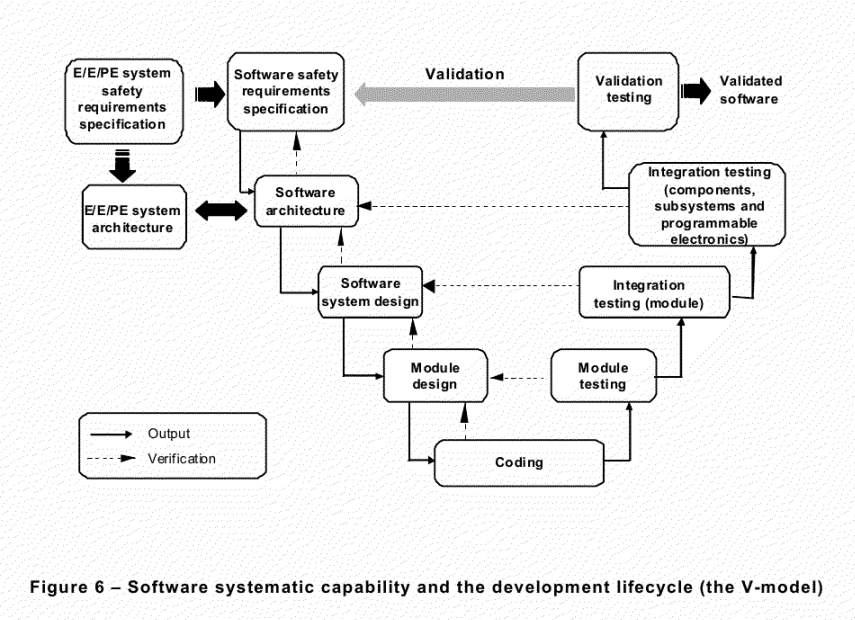
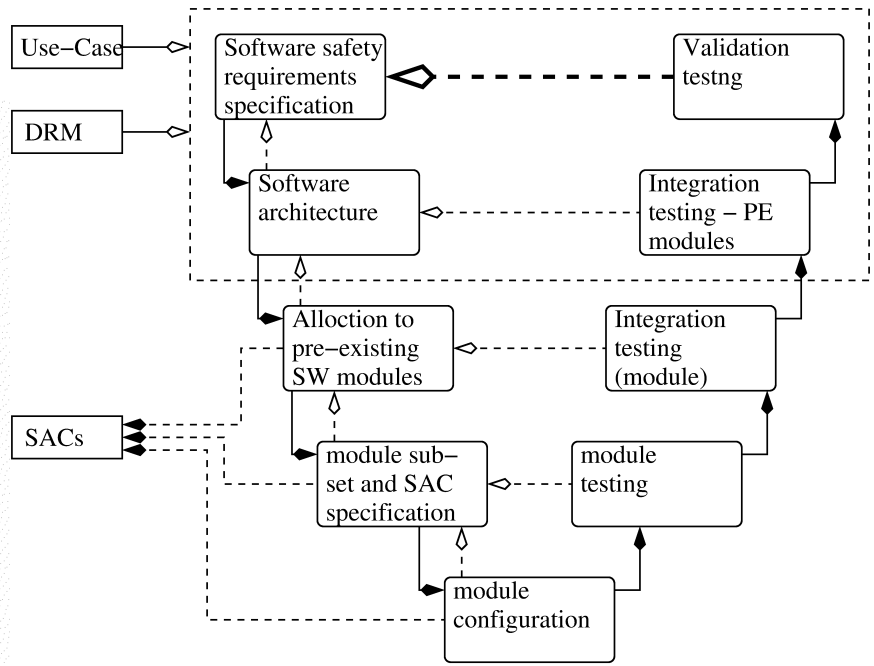
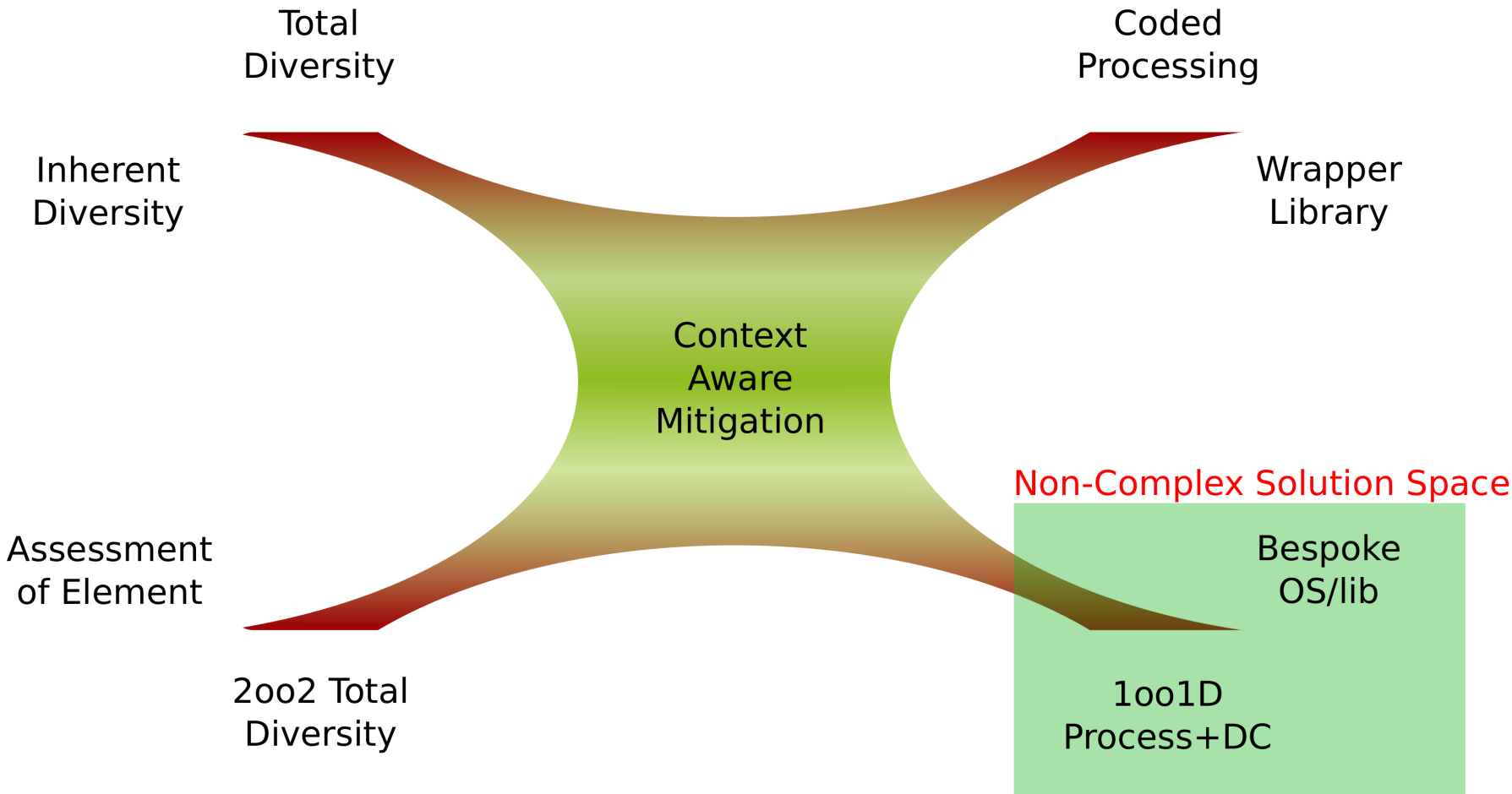


Figure 6 – Software systematic capability and the development lifecycle (the V-model)

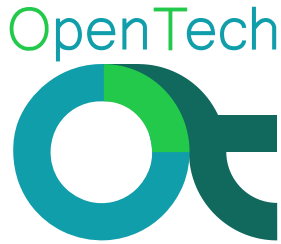


Softwar systematic capability – V–model for pre–existing softwa

Systematic and random fault solution space

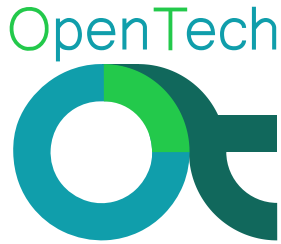


Logical Isolation



- Design-Level Diversity (e.g. SICAS ECC)
- Capitalizing on Security (ASR, SSP, etc).
- Automated Code-level Diversity (pitsfield, coccinelle)
- Inherent Diversity

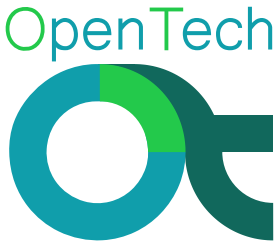
Logical Isolation



- Design-Level Diversity (e.g. SICAS ECC)
- Capitalizing on Security (ASR, SSP, etc).
- Automated Code-level Diversity (pitsfield, coccinelle)
- Inherent Diversity

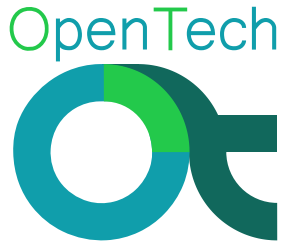
Safety properties need to be in sync with security requirements or we will not build safe systems (61508-1 ED 2 7.4.2.3 -> 62443)

Inherent Diversity



- Multi-cores are highly random hardware
- physical + pseudo concurrency -> nondeterminism
- non-deterministic optimizations - caches, dynamic wait-states, etc.
- deliberate SW/HW randomization for security
- Cross-core calls

Inherent Diversity

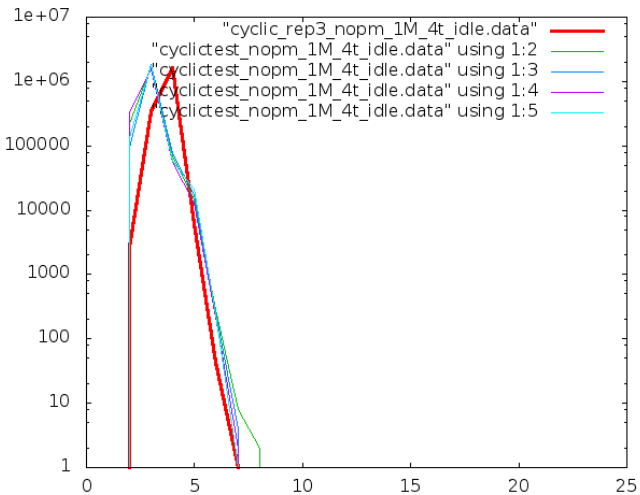


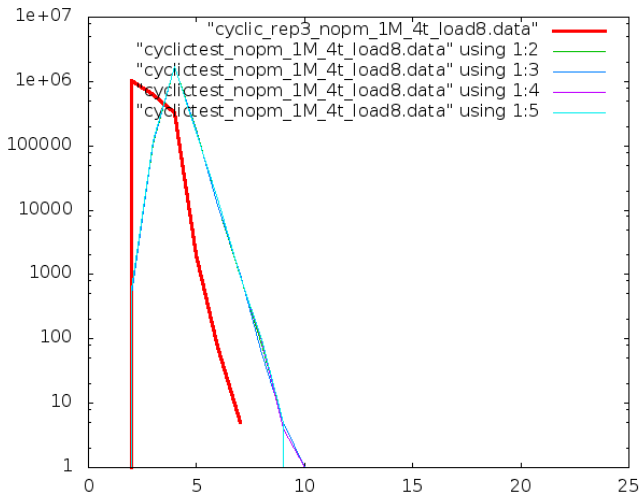
- Multi-cores are highly random hardware
- physical + pseudo concurrency -> nondeterminism
- non-deterministic optimizations - caches, dynamic wait-states, etc.
- deliberate SW/HW randomization for security
- Cross-core calls

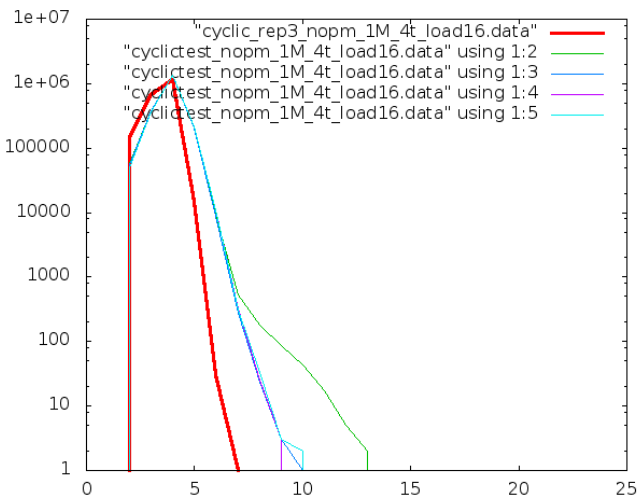
We can try to go on and force system to be deterministic but if we want to build effective, safe systems, we need to learn how to capitalize on complexity and stop fighting it.

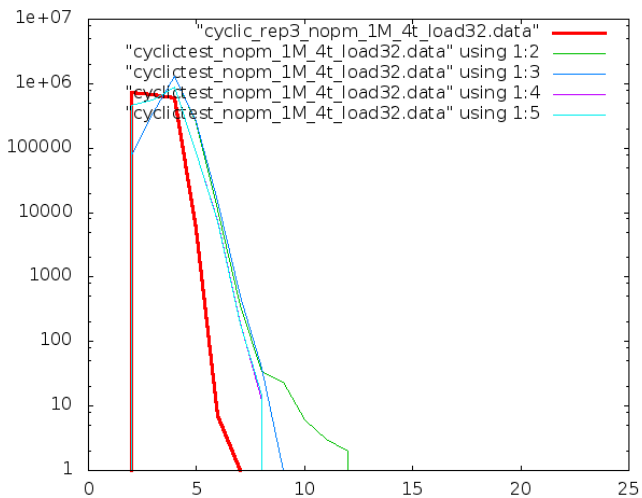
Determinism from randomness ?

i3 4 Threads, load 0,8,16,32 winner model



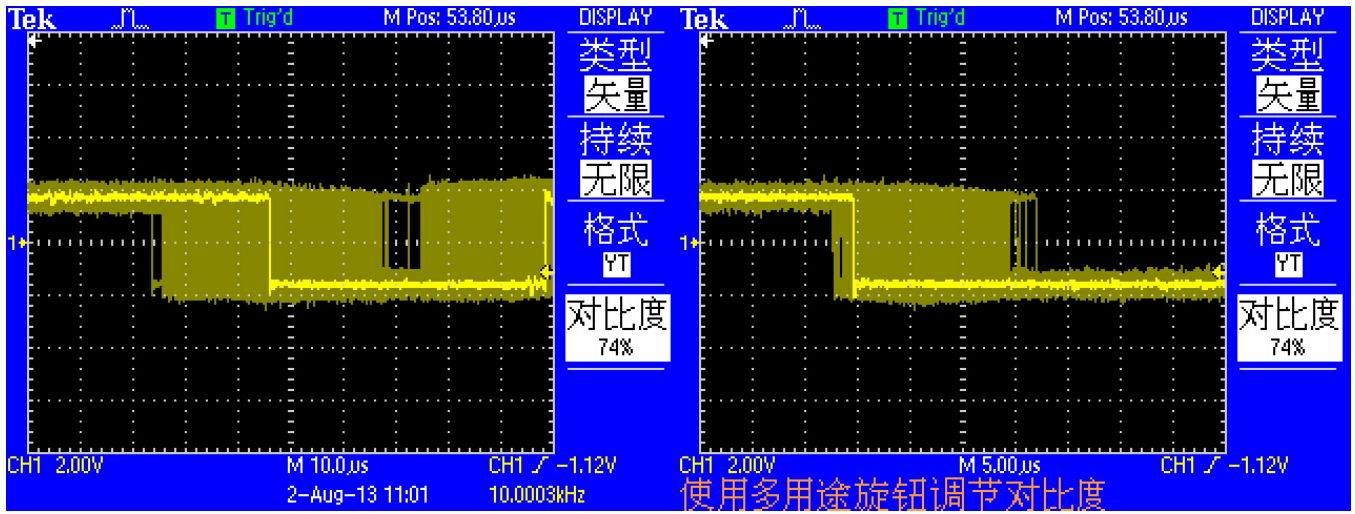


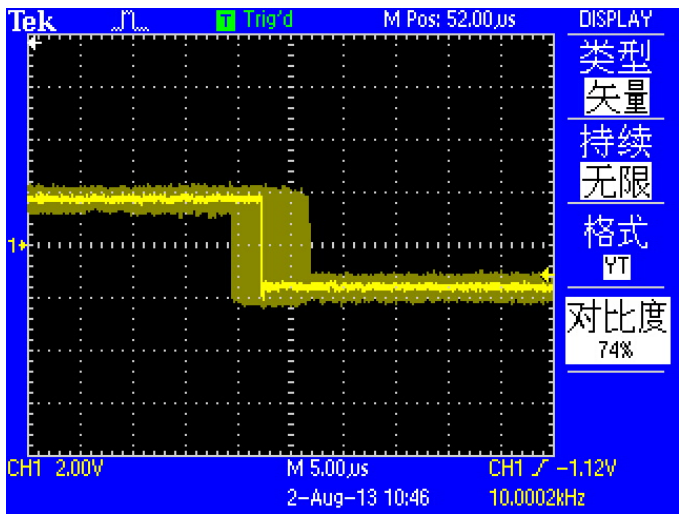




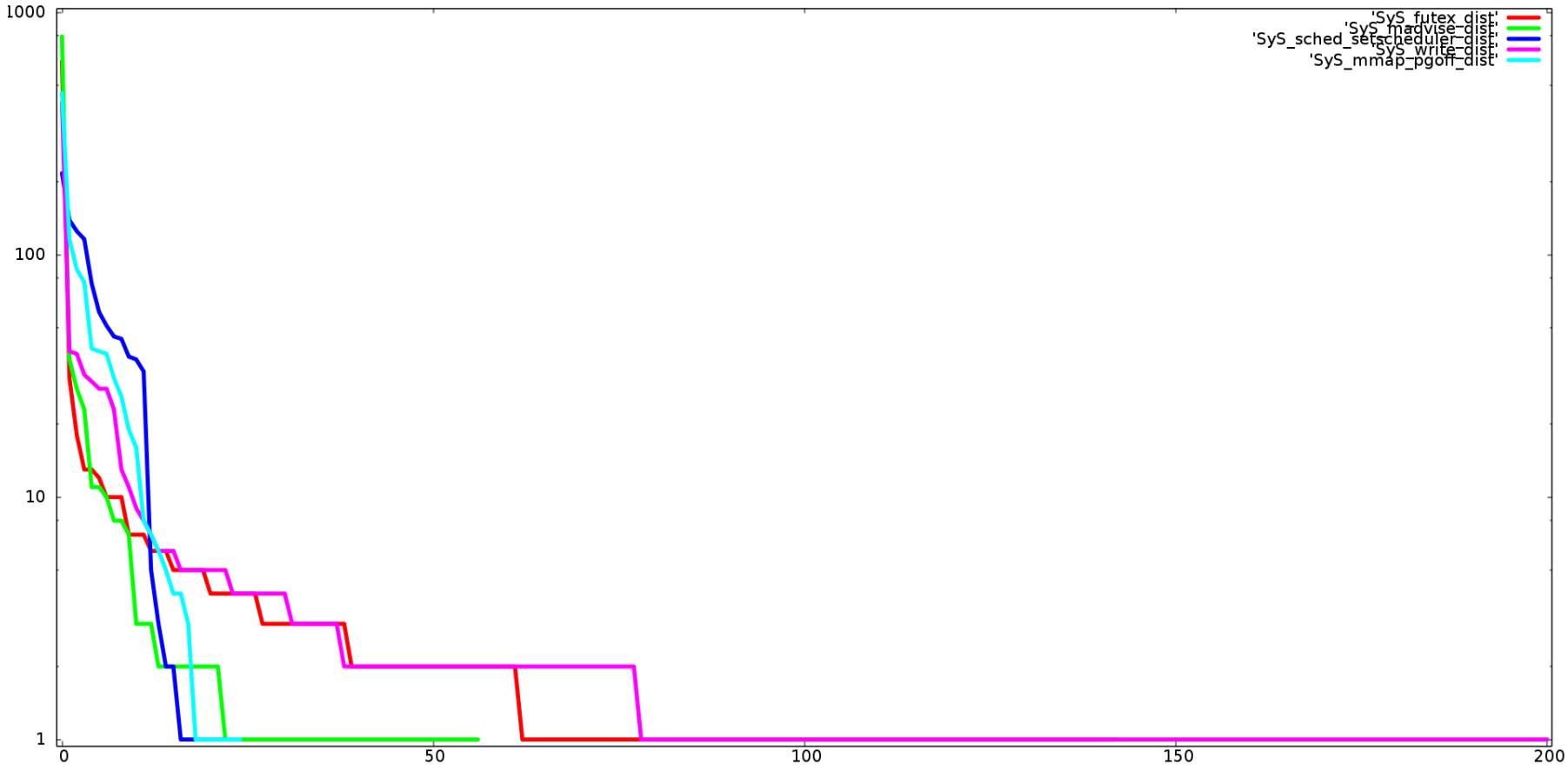
. . . and one more for the HW folks

i3 4 Threads, load 0,8,16,32 idempotent operation

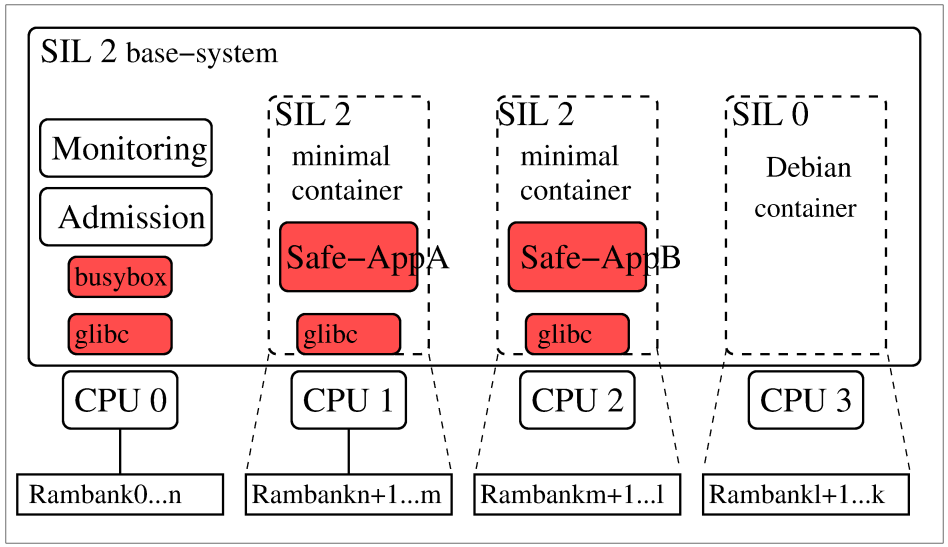




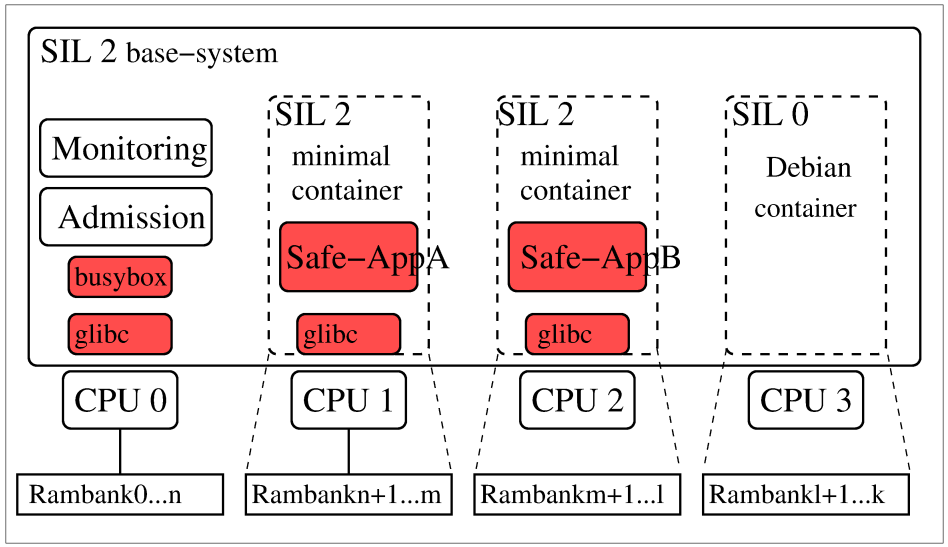
. . . and one more for the HW folks

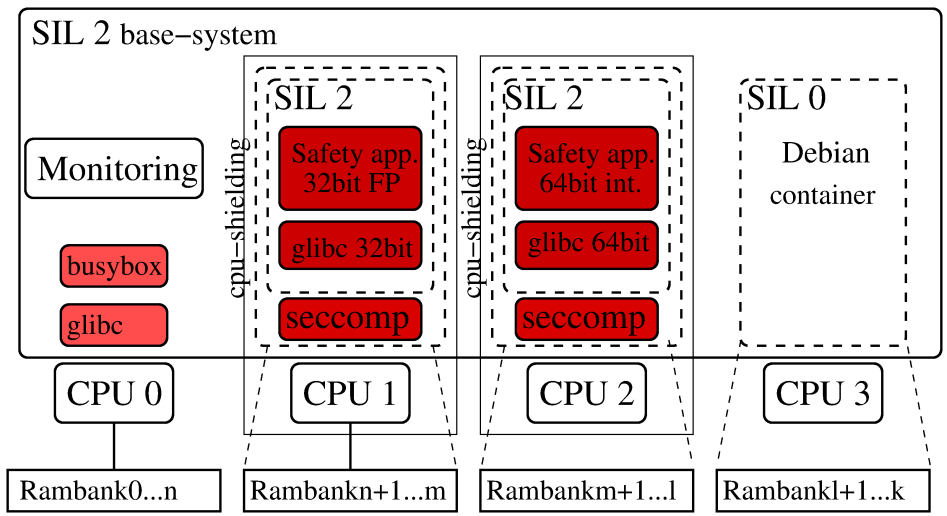


Regaining design space

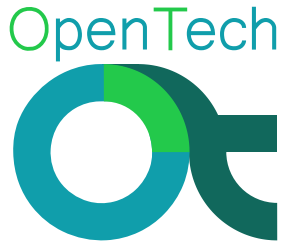


Regaining design space





Conclusions

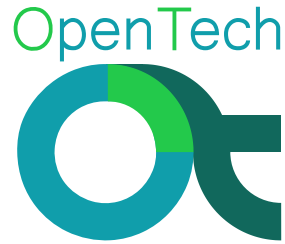


- Safety related systems need to capitalize on complexity.
- Unifying Security and Safety will mandate rethinking methods.
- Utilizing highly-complex HW/SW mandates context aware logical isolation.
- Inherent diversity seems to be an effective and efficient generic logical isolation potential.
- The SIL2LinuxMP architecture allows building diversity based approaches.

SIL2LinuxMP is working on establishing the foundations

http://www.osadl.org/SIL2

Thanks! . . . and btw.

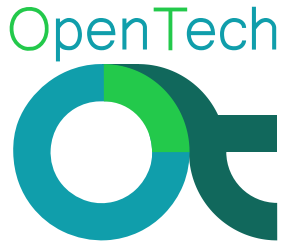


18th Real Time Linux Workshop
and
7th Real Time Linux Summit
October 19 to 21, 2017
at the Czech Technical University in Prague, Czech Republic

Featuring a **Open-Source in Safety** Track!

More Info: https://www.osadl.org/RTLWS

Thanks! . . . and btw.



For those interested in certifying open-source components the open-source way, we invite you to join the

Open Source Development Lab (OSADL)
www.osadl.org

And participate in safety critical working group.

Nicholas Mc Guire <*safety@osadl.org*>