The Road to Safety Certification:

Overcoming Community Challenges to Institutionalise Changes Required for Safety Certification

Lars Kurth

Community Manager, Xen Project

Chairman, Xen Project Advisory Board

IRC lars_kurth

In an earlier presentation I covered

Why Virtualize Embedded Systems

Hardware consolidation, Portability, Flexibility, Cost

Xen and Embedded: A short History

Multiple vendors targeting embedded and safety Use-Cases
Production usage in non-safety and very few in a safety context

The impact on the Xen Project

Functionally a good platform for mixed-criticality workloads
Reference stacks including Xen for automotive
Safety certification needs to be resolved for wider adoption

Safety Certification: A few highlights from our journey

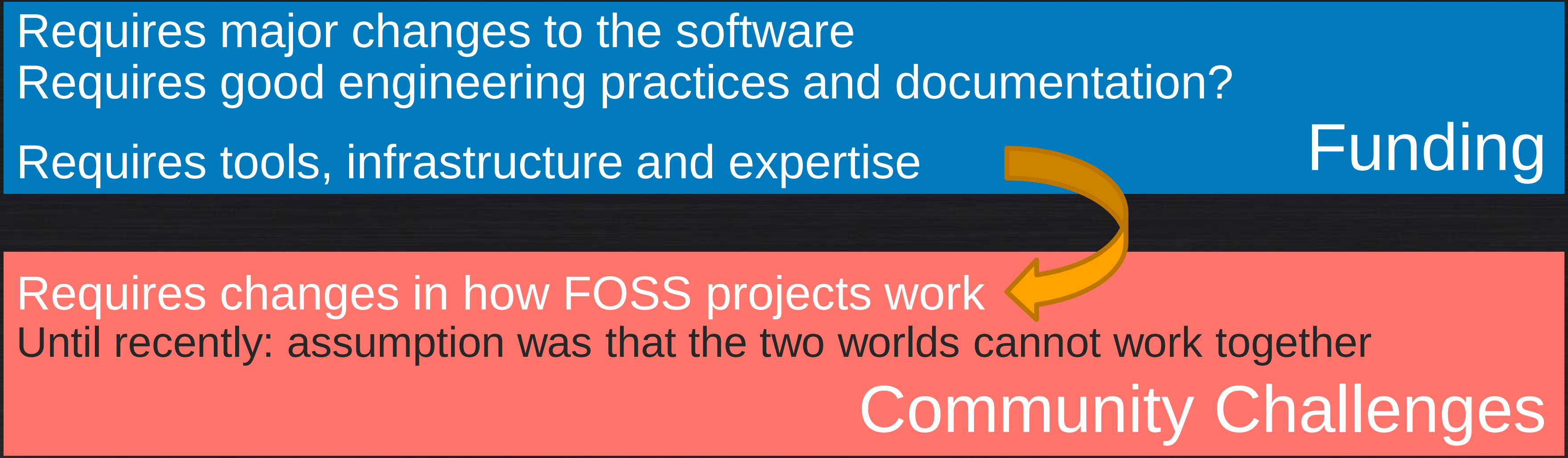
Will cover community aspects in more detail here

static.sched.com/hosted_files/ossalsjp19/45/XenFusa-Overview-converted.pdf

What does it mean to be Safety Certifiable?

Can FOSS SW be used for FuSa?

Yes, but there are many barriers



Requires major changes to the software
Requires good engineering practices and documentation?

Requires tools, infrastructure and expertise

Funding

Requires changes in how FOSS projects work
Until recently: assumption was that the two worlds cannot work together

Community Challenges

At Technical Level

The product requirements are defined

Demonstrate that these are correctly implemented by architecture, unit design, code

- Reviews
- Requirements traceability
- Testing, including measurement of code coverage
- Safety manual and analyses

The requirements, architecture, unit design, code, testing comply to the best practices defined in the safety standards

At Development Process Level

The development process complies with ISO / IEC

- With tailoring: everyone tailors

Safety case: Demonstrates that the process was followed

- Change management (everything is version controlled)
- Process documentation and other standards (reqs, designs, …)
- Project Infrastructure / automation

The more you tailor, the higher the risk that the safety case does not pass and the higher the upfront cost

- Tailoring = funding a specialist consultancy

Verification

Demonstrates that the everything has been done correctly OR argue that what you have done is as good as what the standard requires

Performed by an assessor: need to be confident that Verification will pass, before attempting it

Can only be done if assessors are actively involved in the process or your developers are experienced in FuSa

Verification of Existing Software

You must expect:

- Major re-work of the codebase, including interfaces, modularity, reduction of complexity, …
  - Scale depends on target safety integrity/assurance level
  - And your starting point
- Addition of missing artefacts: specifications, testing, etc.
- To define your development process and extend/modify where there are gaps
- Enforce the development process

Challenges that need to be overcome

Access to Specifications and Expertise

Established developers don’t have a safety background

Could be fixed by training: neither desirable, scalable or indeed necessary
What you need: Sufficient awareness of concepts and terminology

Bringing in new people / developers with relevant expertise

Standards are typically proprietary and complex

MISRA C Standard: licensed to a user @ approx. USD 15
Other standards are more expensive > USD 1000

Significant scope for different interpretations and tailoring

It is absolutely essential that the project has access to specialist expertise

Bridging OSS & FuSa

You need a support infrastructure with experts at hand

Ideally safety certification assessors who can advise key community members how to resolve certain situations → needs to be funded

Needs to be done such that the meritocratic community model is not broken

What if?

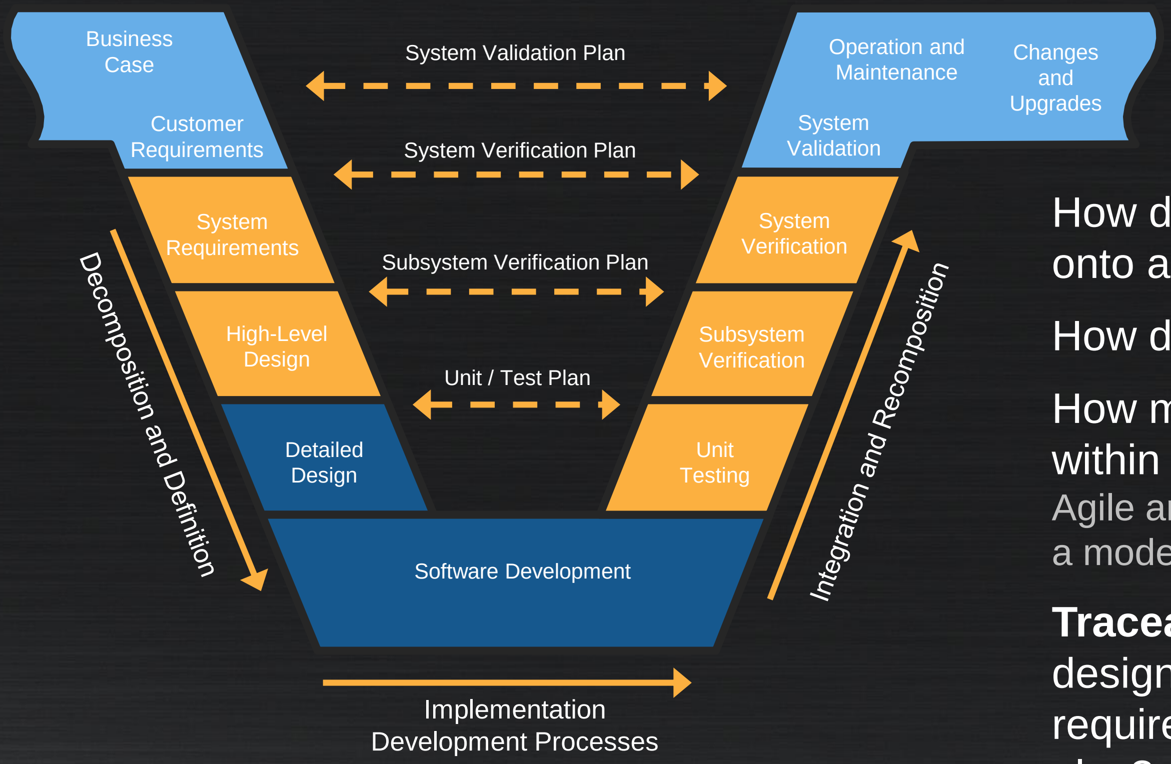
You had developers and companies with FuSa and OSS expertise?

And also with knowledge of the codebase?

We have this in the Xen Project: multiple consultancies (SMEs)

BUT: needs to be funded

Development Process and Traceability



How do you map this onto a FOSS development process?

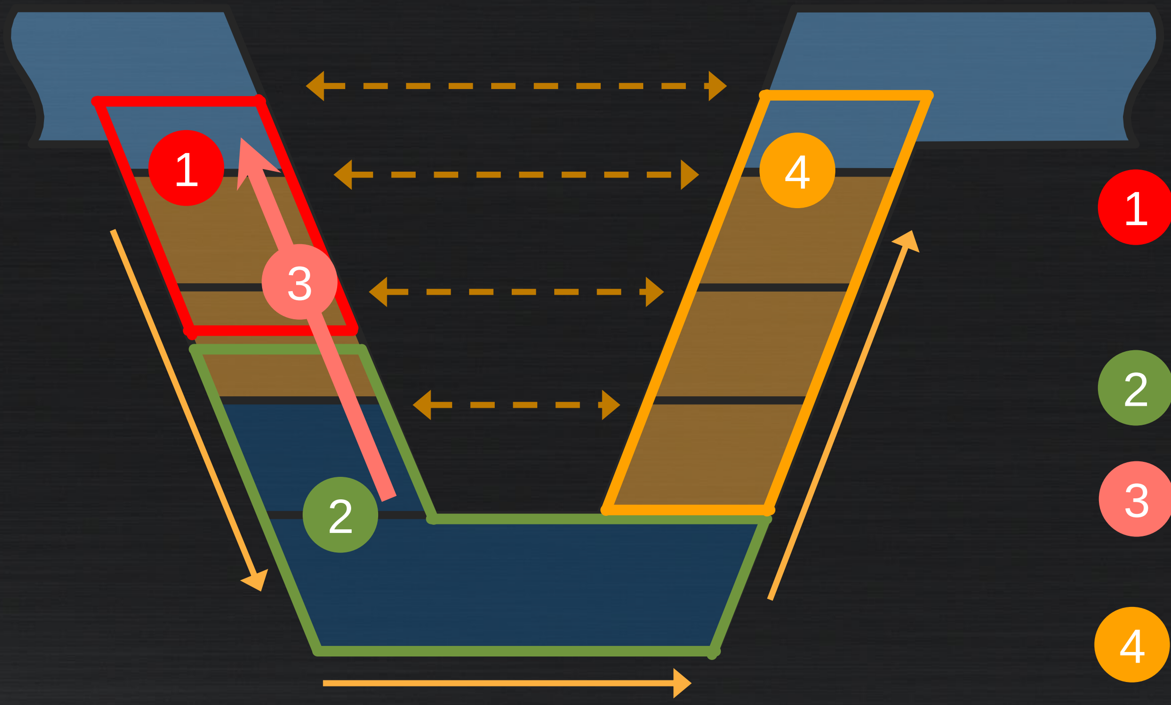
How do you get community buy-in?

How much can be tailored within ISO / IEC ?

Agile and ISO / IEC can provide a model which may fit

Traceability: how do you prove that design and architecture satisfies requirements and tests verify these also?

What you normally have in FOSS is …



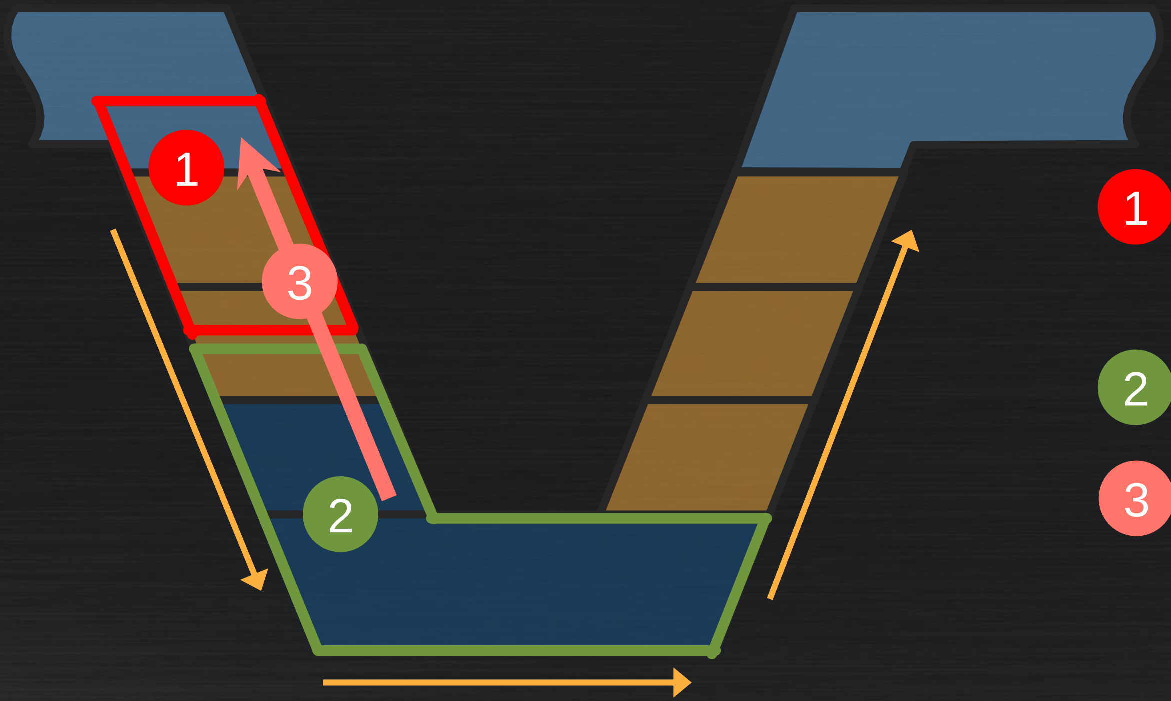
1 Not at at all, or outside
Not a huge effort to retrofit
Valuable for developers & users
Does not change often for a Hypervisor

2 Frequently as good or better
than proprietary. Process discipline

3 Not at all. Difficult to maintain
manually. Should not change that
often

4 A subset of this usually exists, but
typically tests **code**, **not requirements/specifications**.
That’s the most expensive part to
address.

What must be upstream: all key inputs …



1 Not at at all, or outside
Not a huge effort to retrofit
Valuable for developers & users
Does not change often for a Hypervisor

2 Frequently as good or better than proprietary

3 Not at all. Difficult to maintain manually. Should not change that often

Tooling Availability

Compilers, linkers, etc.

Need to be certified – typically proprietary

Such tools would need to be integrated into a CI gate

In essence this means buying licenses and/or partnering with vendors

Coding standard compliance

Compliance checking tools for MISRA C Standard – typically proprietary

There are some FOSS tools, which check subsets of the standard

Again: needs CI integration and licenses and/or partnering with vendors

Traceability

Proof that tests satisfy requirements (and vice versa)

Linkage between requirements and specifications (and vice versa)

Commercial software is expensive and does not fit into an open source workflow

Only 1 active project which does some of what is needed: Doorstop project

Coding Standards: MISRA C

Required by most safety standards

Misra C is a de-facto standard
10 Mandatory, 111 Required and 38 Advisory rules
Required rules depend on certification level: can be deviated from

Community Challenges

Proprietary spec and tooling
Coding guidelines and checking (e.g. via CI)
How to avoid unnecessary discussion, while recognizing valid concerns
How to deal with changes with high code churn
(e.g. past supported releases and backporting of security fixes)

Fast growing FOSS projects are user adoption driven

The extra cost of safety certification is significant

The risk that upfront investment doesn’t deliver is very high!

FuSa breaks the traditional OSS growth model!

Funding of FuSa

How do we fund access to development tools and expertise?
How do you fund filling the “gaps”?

ELISA Project

Founding members: Arm, BMW Car IT GmbH, KUKA, Linutronix, Toyota
Reduce risk by providing tools, processes and patterns

Zephyr

Appears to be funded by Intel and some partners to establish Zephyr as a safety certifiable open source RTOS

Xen Project

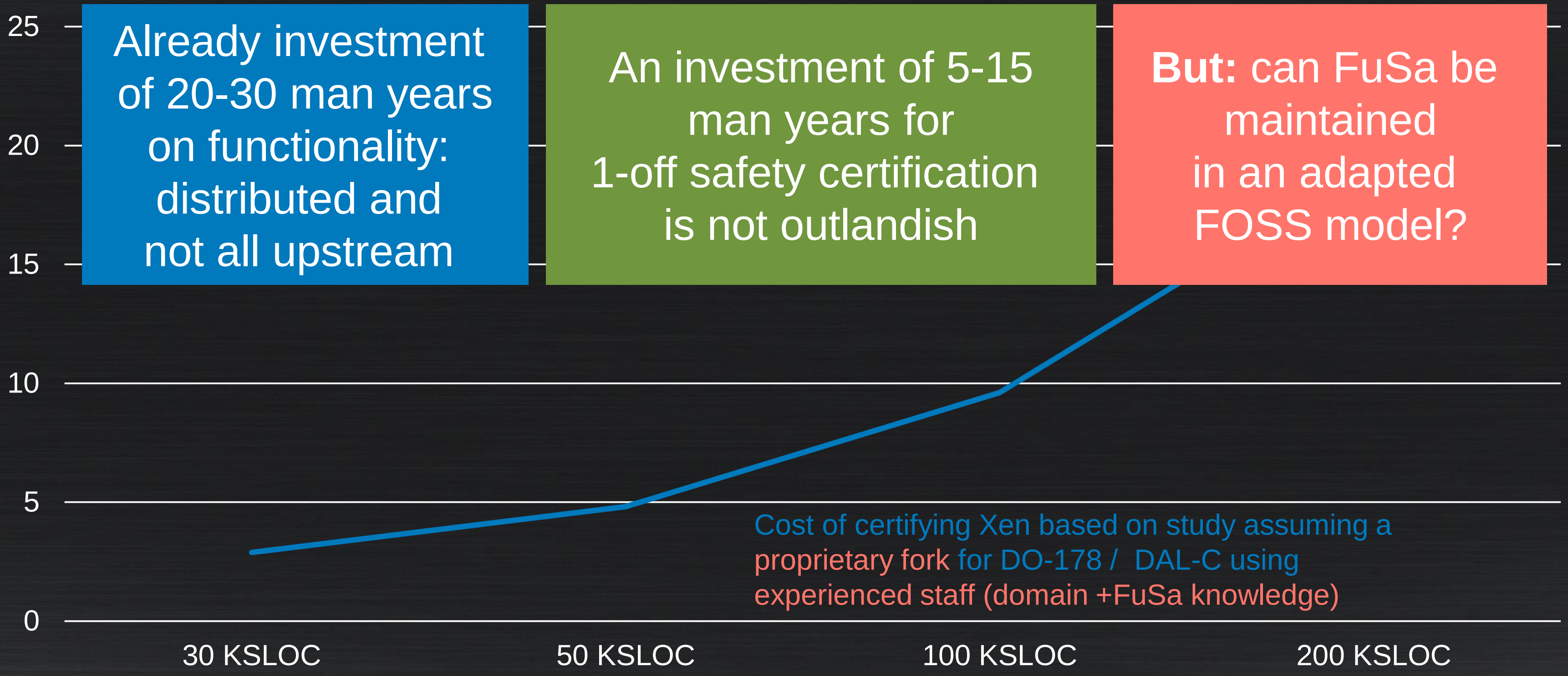
So far on a per contributor basis from various organizations that have a vested in safety certifying Xen. Possible partnerships with assessors and tooling companies (in progress).

Ultimately this is not going to be enough: approach is to make progress in some areas to demonstrate progress with reference implementations and unlock further funding.

Safety Certification in Xen Project

Establish the Feasibility

Minimal Xen on Arm: costs for FuSa



Already investment of 20-30 man years on functionality: distributed and not all upstream

An investment of 5-15 man years for 1-off safety certification is not outlandish

But: can FuSa be maintained in an adapted FOSS model?

Cost of certifying Xen based on study assuming a proprietary fork for DO-178 / DAL-C using experienced staff (domain +FuSa knowledge)

How we have approached safety to date

2012 – now: Xen commercial distros with some support for safety
BUT: no upstream support, no community engagement

2016 – now (and for the foreseeable future)
Technical: develop and upstream functionality needed for mixed-criticality workloads

2019: Start a process to
establish feasibility and to create a plan

2019: Planning - WIP
Agreements, Funding, Plan, Risks

2019: Create Enablers - Plans
Infra, Tools, Community

Bring together Industry and Community

2 day workshop in March 2019 with 25 attendees

Community Reps and Support

Project leadership team (except for 2)

Kate Stewart as observer / advisor

THE LINUX FOUNDATION

Safety Assessors

RESILTECH

exida®

TÜV SÜD ISO 9001

A HORIBA COMPANY MIRA

Vendors with investment in Xen

arm

<epam>

RENESAS

XILINX

Vendors with product interest / skills

ADIT

STAR LAB SOFTWARE

PERFORCE

Objectives

Create an understanding between the community and industry

Terminology, Concepts, etc.

How safety certification works: look at different standards, routes, requirements

Explain assets and processes

Establish “red lines”

Principles the community can agree to or would object to

What level of change would be acceptable

Identify potential obstacles

Establish whether Xen Project is safety certifiable

If so, create a candidate set of feasible certification routes

Establish a rough action plan on how to progress

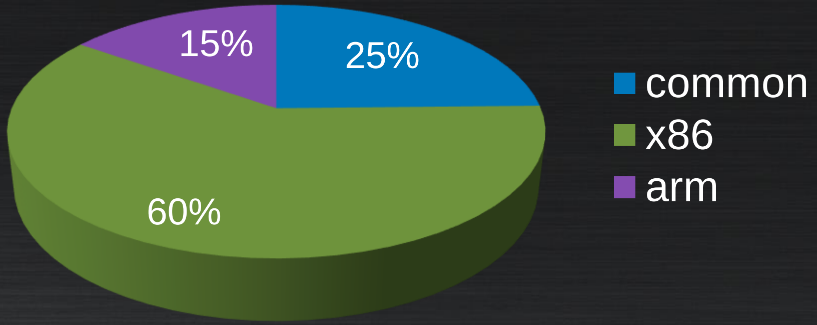
Xen Project is Somewhat Different

Compared to Zephyr and ACRN it has an established user base in server, cloud and security applications on x86

Contributor community is **diverse**

Xen on x86 is not really suitable today for embedded/mixed-criticality

But Xen on Arm is: but was originally designed for servers
Effort to redesign and refresh for mixed-criticality use-cases is scoped and sized
Implementation and funding WIP → opening to make this “safety-friendly”



Anything we do for safety can only be done if there is agreement to implement changes in a relevant subset of common code

High Level Agreements

Split development model with an open and a closed part

Everything that is valuable to the wider community in the open part,
e.g. documentation, tests (not all of them), traceability, automation and infrastructure,….

Everything that creates code churn if it wasn’t open as much as possible:
e.g. coding standards (MISRA)

Changes to the open development workflow are minimal

There must be a benefit the community (including for common code)
Broad agreement that codified requirements, more designs, more tests, traceability information are all beneficial for the project as a whole

BUT: the workflow is git centric and there should not be no parallel universe of additional infrastructure and tools outside of git

- Requirements, specs, etc. must all be stored in tree and covered by the projects review workflow
- Traceability reports, etc. must be generated from in-tree artefacts

Red Lines

Filling the gaps

Gaps in terms of documentation, specifications, safety manual must be developed and contributed by vendors interested in safety.

Tests can be proprietary, if there is a 3rd party CI integration and commitment to triage and fix issues upstream (similar to what OpenStack does)

There must be investment in necessary project infrastructure to enable this.

Contributions have to be reviewed to go into mainline: there must be a commitment to "build new maintainers" (by above vendors performing code reviews)

Maintaining

Vendors will need to step up with maintainership, code reviews, test triage, supporting the new infrastructure, …

Otherwise: all the initial work will become stale and will create burden for everyone else

Accountability for the Implementation

You might have the coolest open-source project with a super complete feature-matrix that is safety-certifiable

No-one will use it unless there is a clearly identified entity that is responsible for the safety sign-off for that project

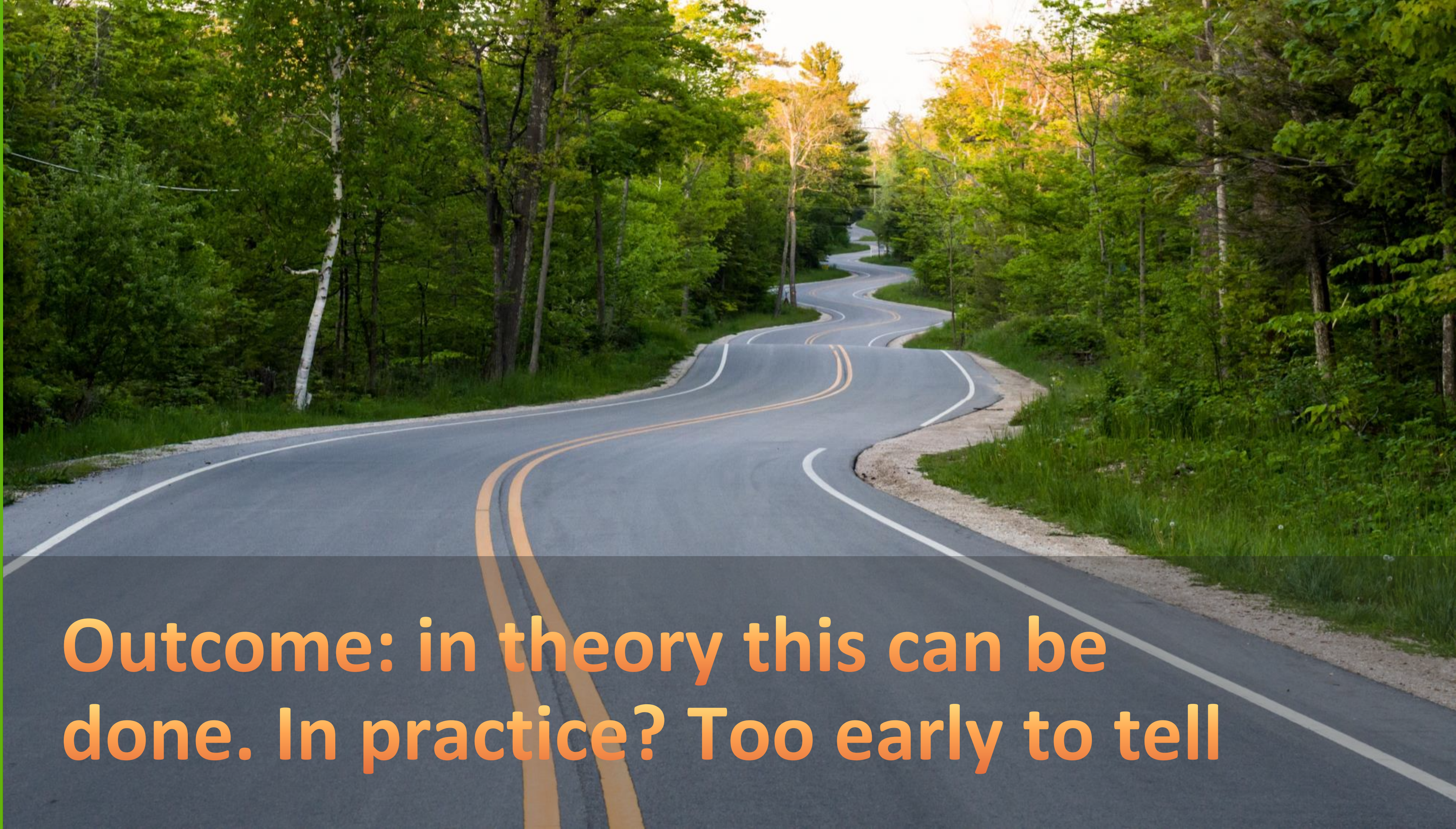
In the Split Development Model this can be done by

- A commercial entity which is accountable: either a single vendor, multiple vendors or a group/consortium that collaborates with the community
- Projects such as ELISA are also looking at this

Reference Stacks

Create reference stacks for safety use-cases supported by different vendors and eco-systems

- Already have the EPAM automotive stack
- Have a XILINX mixed-criticality stack
- Another one in the pipeline (under NDA)
- Others are being discussed/proposed by groups that previously were not engaged with Xen



Outcome: in theory this can be done. In practice? Too early to tell

FuSa SIG with Workstreams

Subgroups meet at least every other week. Partly resourced

Community Reps

Lars Kurth (chair and project mgmt)
George Dunlap (committers)

Assessors

RESILTECH

exida®

MIRA
A HORIBA COMPANY

TÜV SÜD
ISO 9001

Stream Owners and Implementers

Lars Kurth

THE LINUX FOUNDATION

<epam>

XILINX

RESILTECH

arm

RENESAS

Other Members

ADIT

The next stage is VERY Work-in-Progress

Activities

Certification scope route and overall plan and strategy

A set of very early drafts: still in bootstrap mode

Following an agile approach

Starting to break down dependencies and priorities

Funding and Resourcing

Some secured

More needed

Some ideas around business models/research grants for funding

Possibly additional SIG members volunteering time and resources

Work Streams 1/3

Safety Management System (for the closed part)

Resourced to create plan/strategy
Must be designed to co-exist with Xen mainline development

Documentation

Draft strategy (not yet published)

- Around inputs into certification process (Requirements, Specs, API docs, …)
- A set of leads for in-code, in-tree encoding
- Ideas for traceability, which need to be verified
- Some can live in closed part

Some Xen vendors have content that could be used as seed
Better dev docs is what committers want and support

Work Streams 2/3

Verification Tests

Focusing on CI capability vision and implementation first (CI v2 and v3)

Some is resourced and aligns with plans the project already had

Something the community wants and agreed to last week

Capacity issue with traditional CIs that test on lots of different HW

- Can’t integrate CI **before start of code review**
  - Too expensive to purchase HW and to maintain HW to enable needed scale
  - Issue: can’t test EVERY merge request
- Front-load the review process with additional CI capability
- Use automation bots as much as possible
- For e-mail based code-review
  - Recently patchew, patchwork, lore have improved

Work Streams 3/3

Community Interactions and Processes

Focusing on using FuSa to help address long-standing problems

- With funding and resources

This seems to work for now: the devil will be in the detail

MISRA C compliance:
needs planning and a process to find a compromise

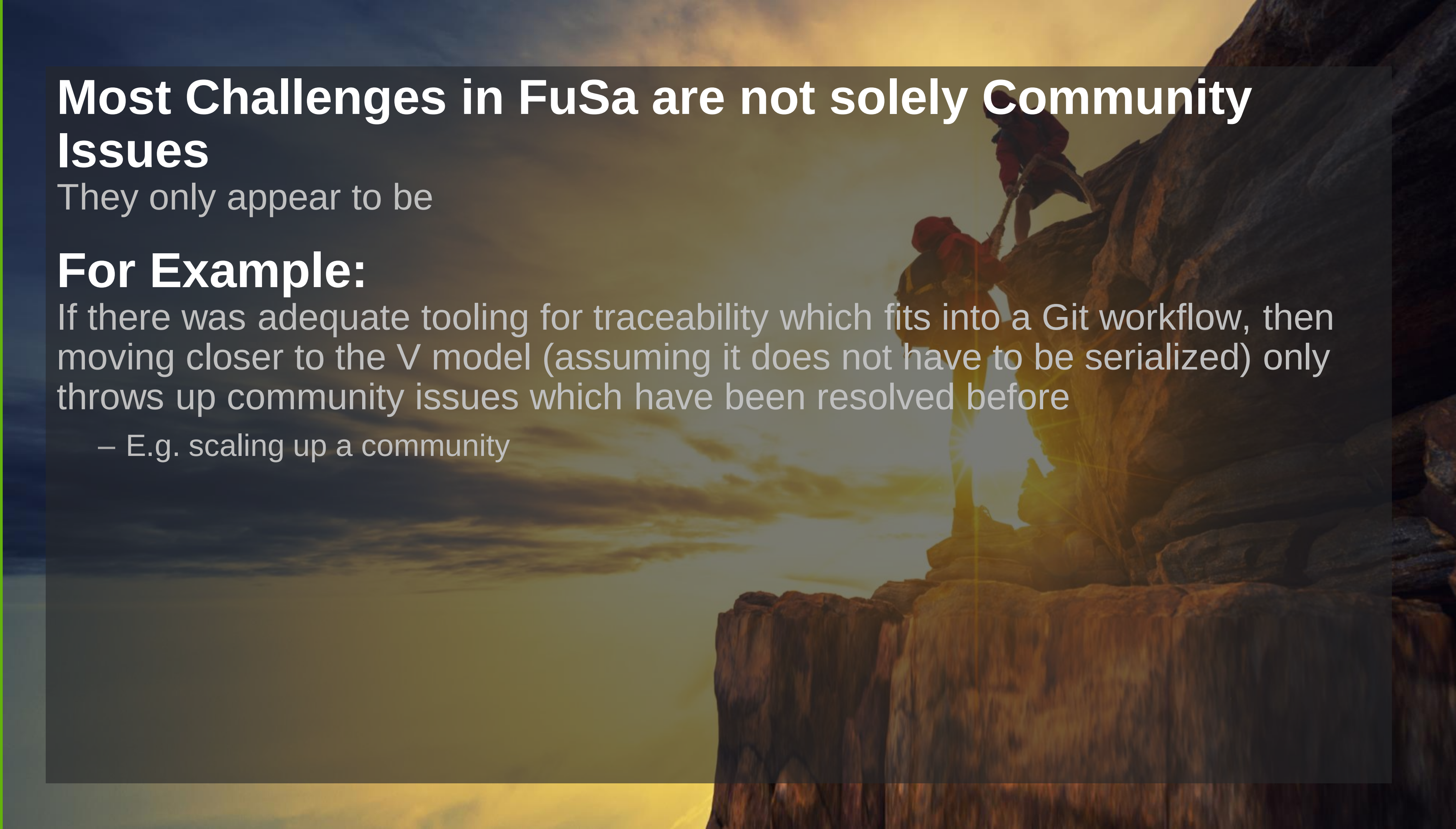
Process Automation Tools

Surveying what is available, usable and a good baseline to extend

- Dependency on ELISA Project

MISRA C: looking at Perforce QA Verify, Bugseng Éclair and cppcheck

How to solve community Challenges for FuSa?



Most Challenges in FuSa are not solely Community Issues

They only appear to be

For Example:

If there was adequate tooling for traceability which fits into a Git workflow, then moving closer to the V model (assuming it does not have to be serialized) only throws up community issues which have been resolved before

- E.g. scaling up a community

Questions

Picture by Lars Kurth