



October 2026

Preparing Xen for Functional Safety

Lessons From a Mature Open Source Project

Cody Zuschlag



xenproject.org



DRIVING VIRTUALIZATION FORWARD



Why Xen?

I thought we already settled this.



Why Xen?

I thought we already settled this.

Open source, Type-1
hypervisor



Why Xen?

I thought we already settled this.

Open source, Type-1
hypervisor

Designed for mixed OS
/ embedded systems



Why Xen?

I thought we already settled this.

Open source, Type-1
hypervisor

Designed for mixed OS
/ embedded systems

Already part of the AGL
SDV story





Why open source?

Commercial hypervisor

Vendor roadmap

Licensing surprises

Black box

Feature request

Supplier dependency

Open source hypervisor

Your roadmap

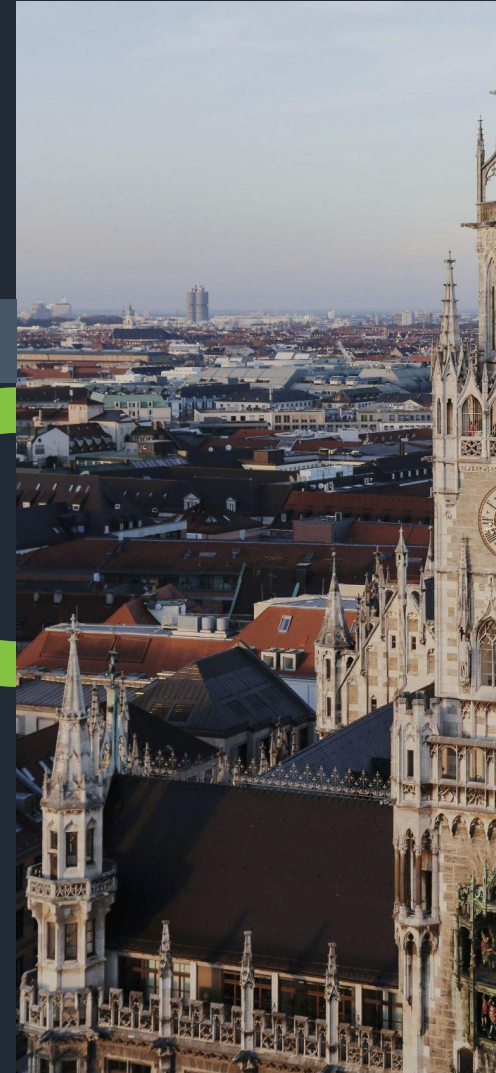
Predictable access

Inspectable

Contribution

Shared ecosystem

Also: your executives already said "open source first," right? 😏



Backed by **automotive, cloud,
semiconductor, and safety-critical
organizations**



AMD

arm

aws

BOEING

<epam>

Ford

HONDA
The Power of Dreams

RENESAS

VATES

XenServer

Xen Safety



1. FuSa Working Group



Xen Safety



1. FuSa Working Group

2. Safety Committee

Xen Project Launches Shared Functional Initiative for Safety-Critical Systems

THE LINUX FOUNDATION | 17 AUGUST 2026



ANNOUNCEMENT

Xen Project Launches Shared Functional Initiative for Safety-Critical Systems



The Xen Safety Committee gives organizations a collaborative path to develop and maintain reusable engineering artifacts for functional safety certification

Summary

- The Xen Project, an open source virtualization platform hosted by the Linux Foundation, today announced the Xen Safety Committee, a collaborative initiative for developing and maintaining reusable engineering artifacts that organizations can use to support their own functional safety certification programs.

How do you make a 20+ year project ready for functional safety?



Working software is not
the same thing as
safety evidence.

Does it work?



Working software is not the same thing as safety evidence.

~~Does it work?~~

Can we demonstrate why it works?



**The inconvenient
truth:**
Xen already exists.



The inconvenient truth: **Xen already exists.**

20+ years

millions of lines of code

multiple architectures

downstreams



existing maintainers

stable releases

historical design decisions

The inconvenient truth: **Xen already exists.**

The challenge:

- requirements may be implicit
- architecture knowledge may live in maintainers' heads
- tests prove behavior but may not map cleanly to requirements
- processes evolved organically
- you cannot throw away a functioning community model

20+ years

millions of lines of code

multiple architectures

downstreams



existing maintainers

stable releases

historical design decisions

Don't start over.
Discover **what you already have**



Don't start over. Discover **what you already have**

Already strong

Needs strengthening

New safety work

Review

CI

Security Process

Stable branches



Don't start over. Discover **what you already have**

Already strong	Needs strengthening	New safety work
Review	Documentation	
CI	Traceability	
Security Process	Coverage	
Stable branches	Reproducibility	

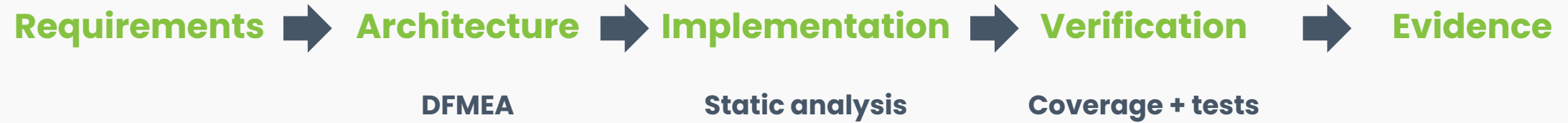


Don't start over. Discover **what you already have**

Already strong	Needs strengthening	New safety work
Review	Documentation	Requirements
CI	Traceability	DFMEA
Security Process	Coverage	Safety evidence
Stable branches	Reproducibility	Artifact lifecycle



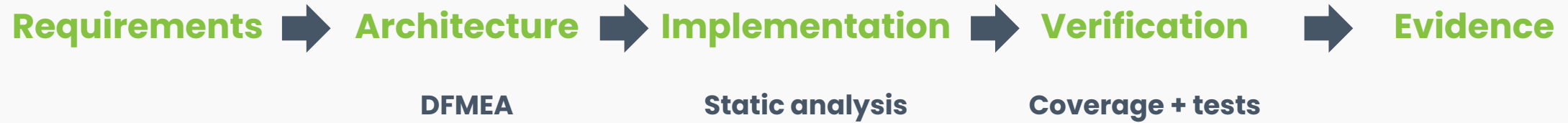
Code is only one artifact



Code is only one artifact

Requirements are being upstreamed alongside Xen

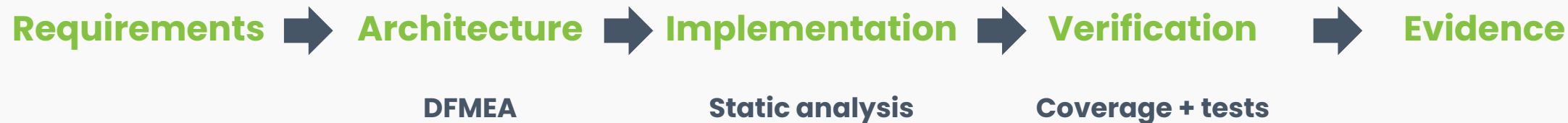
Developed collaboratively across AMD, Arm, EPAM and Renesas.



Code is only one artifact

Requirements are being upstreamed alongside Xen

Developed collaboratively across AMD, Arm, EPAM and Renesas.



The safety work isn't just changing Xen.

It's maintaining the engineering story around Xen.

The artifacts have to move with the code.

Patch



Review



Release



Stable fixes



Security fixes



The artifacts have to move with the code.

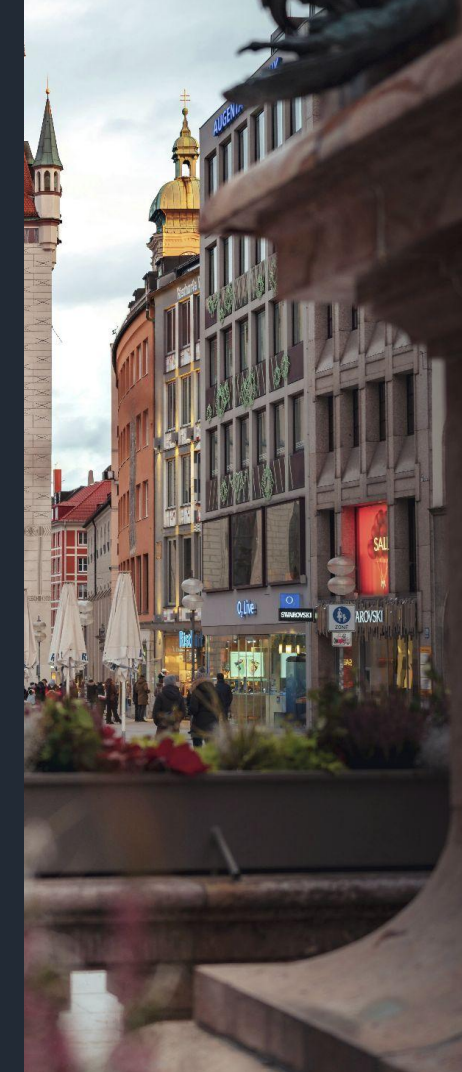


Safety is a community problem

Shared upstream safety foundation

- **AMD, Arm, EPAM, Renesas** → requirements
- **Boeing** → requirements, unit design, unit tests, Ceedling
- **Wider Xen community** → self-test framework

A lot of companies need the same foundational work. Open source gives us a way to stop doing all of it independently.



What other projects can **learn**



What other projects can **learn**

Start with what already works.

Don't rebuild mature engineering processes just to make them look like safety processes.



What other projects can **learn**

Start with what already works.

Don't rebuild mature engineering processes just to make them look like safety processes.

Make implicit knowledge explicit.

Architecture, assumptions and requirements have to leave people's heads.



What other projects can **learn**

Start with what already works.

Don't rebuild mature engineering processes just to make them look like safety processes.

Make implicit knowledge explicit.

Architecture, assumptions and requirements have to leave people's heads.

Treat artifacts like source code.

Version, review and maintain them as the project changes.



What other projects can **learn**

Start with what already works.

Don't rebuild mature engineering processes just to make them look like safety processes.

Make implicit knowledge explicit.

Architecture, assumptions and requirements have to leave people's heads.

Treat artifacts like source code.

Version, review and maintain them as the project changes.

Do shared work upstream.

Keep reusable safety engineering close to the code instead of recreating it downstream.



You don't make a mature open-source project ready for safety by starting over.

You don't make a mature open-source project ready for safety by starting over.

You make its engineering visible, traceable and maintainable.

You don't make a mature open-source project ready for safety by starting over.

You make its engineering visible, traceable and maintainable.

And you build a community capable of keeping it that way.

You don't make a mature open-source project ready for safety by starting over.

You make its engineering visible, traceable and maintainable.

And you build a community capable of keeping it that way.

Come build the shared safety foundation with us.



Any questions ?

Thank you for your attention



xenproject.org