The Road to Safety Certification:

How the Xen Project is Making Progress within the Auto Industry and Beyond

Lars Kurth
Community Manager, Xen Project
Chairman, Xen Project Advisory Board

lars_kurth

Why Virtualize in Embedded Systems?

Consolidation

Reduce cost, size, weight and power consumption

Reduce development costs: platform independence

Security and Safety

Support mixed criticality compositions

(Apps with differing safety, security & real-time requirements)

Safety Certification of the Hypervisor

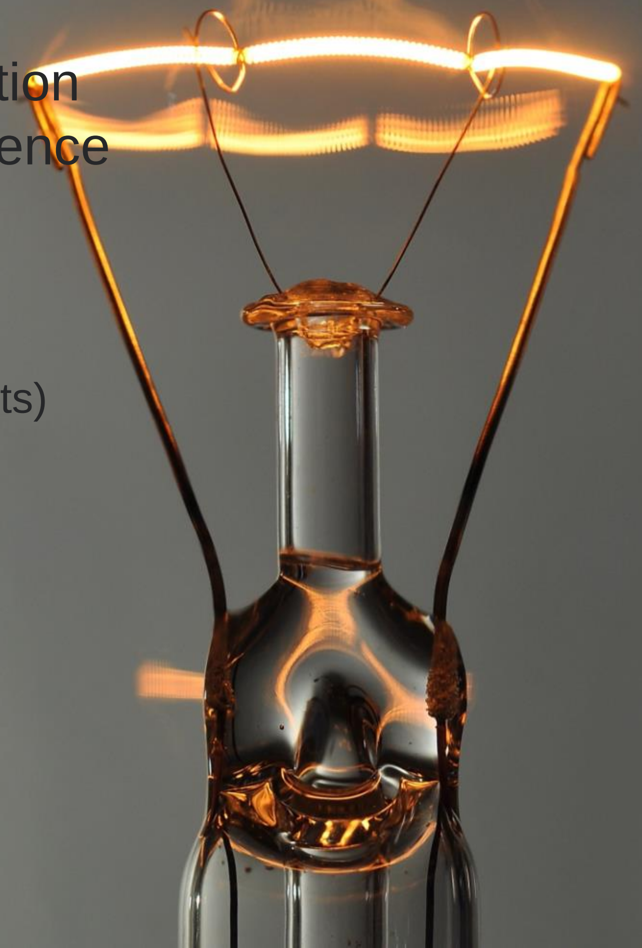
Embedded Requirements

Minimal IRQ latency

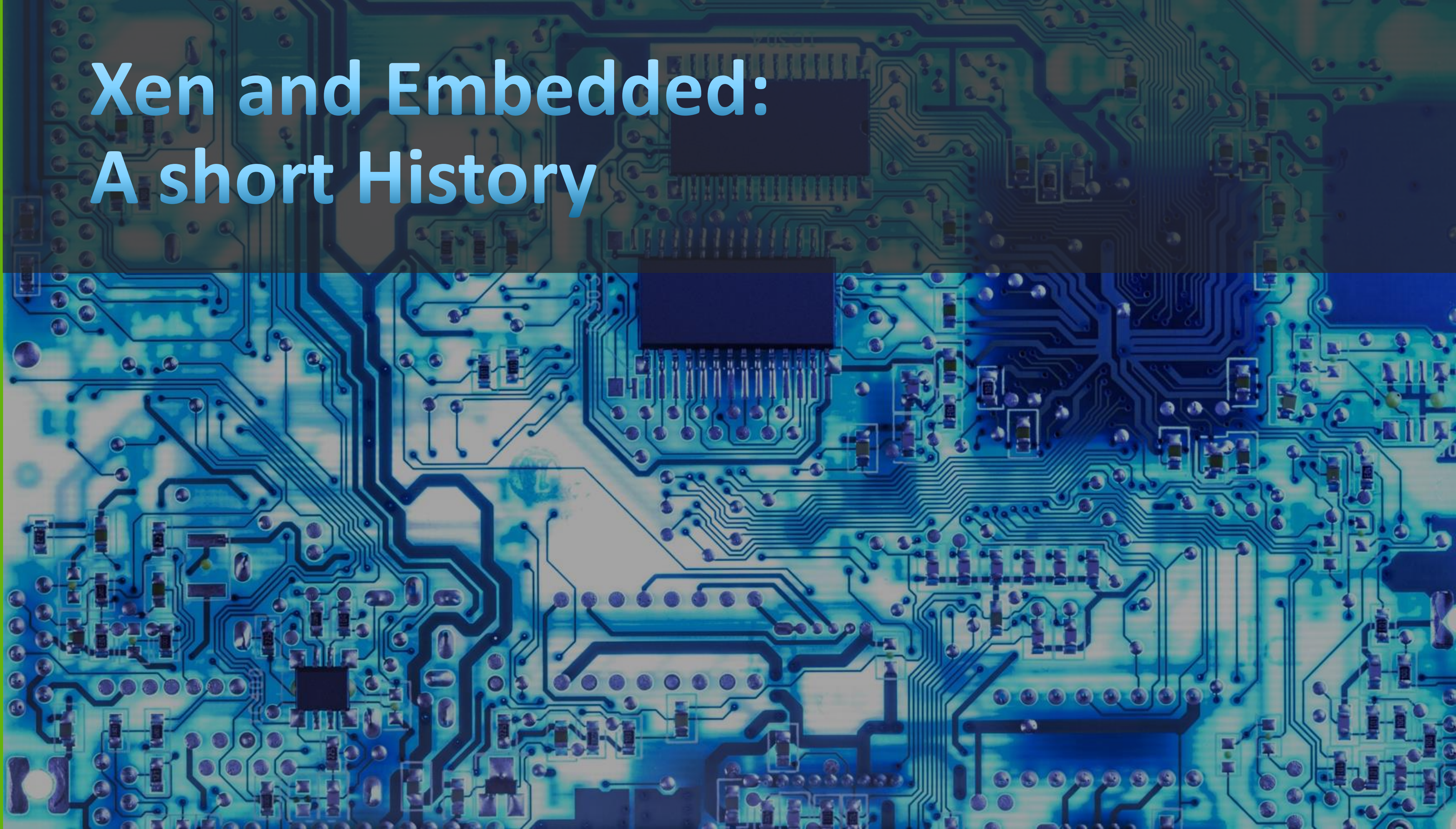
Low or 0 scheduling overhead

Drivers for special I/O devices

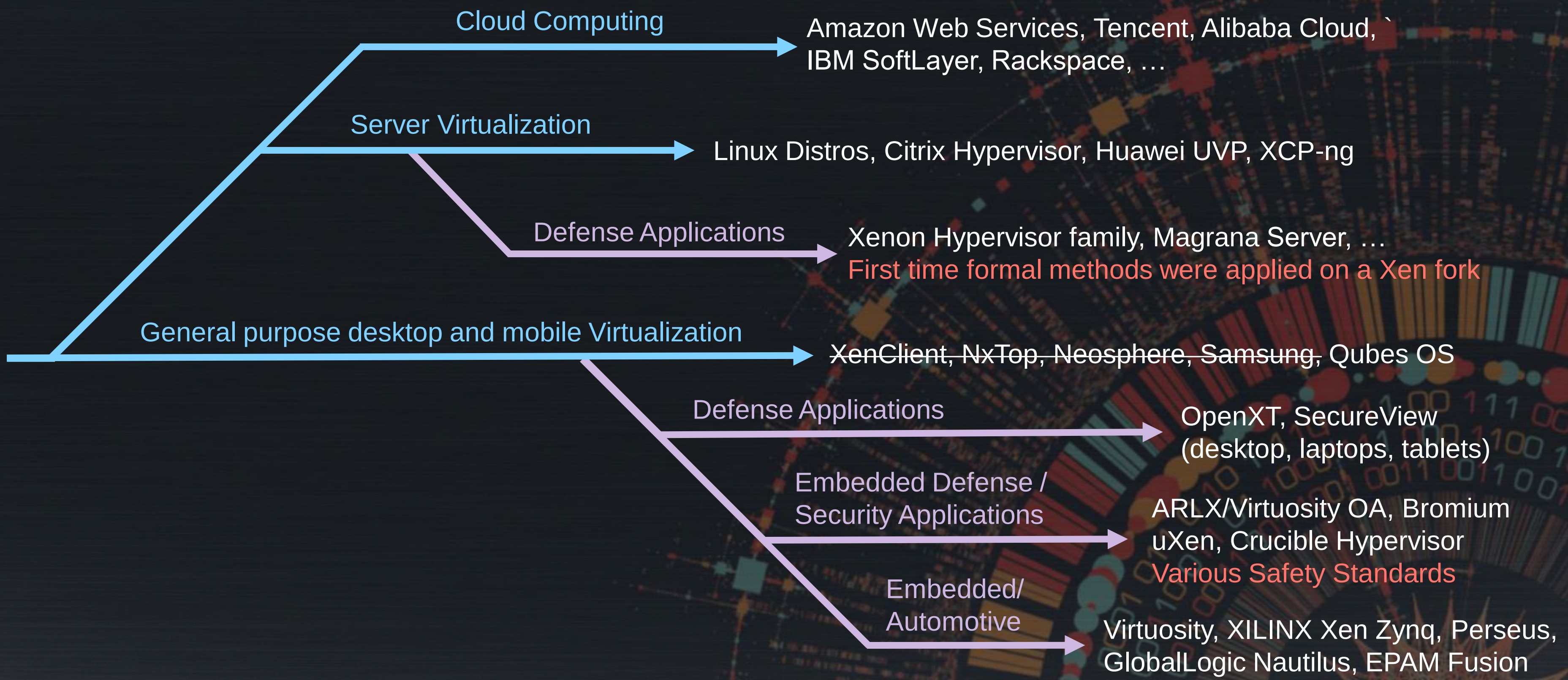
Flexible architecture



Xen and Embedded:
A short History



Xen Ideas/Product Genealogy



Cloud Computing

Amazon Web Services, Tencent, Alibaba Cloud, `
IBM SoftLayer, Rackspace, …

Server Virtualization

Linux Distros, Citrix Hypervisor, Huawei UVP, XCP-ng

Defense Applications

Xenon Hypervisor family, Magrana Server, …
First time formal methods were applied on a Xen fork

General purpose desktop and mobile Virtualization

~~XenClient, NxTop, Neosphere, Samsung,~~ Qubes OS

Defense Applications

OpenXT, SecureView
(desktop, laptops, tablets)

Embedded Defense /
Security Applications

ARLX/Virtuosity OA, Bromium
uXen, Crucible Hypervisor
Various Safety Standards

Embedded/
Automotive

Virtuosity, XILINX Xen Zynq, Perseus,
GlobalLogic Nautilus, EPAM Fusion

2012

Xenon Separation VMM family, CC EAL5+

Fork of cut down version of Xen Project used by the US military.
Certified to CC EAL 5+ (Semiformally Designed and Tested which has some similarity to safety standards). Tracks upstream and maintained with an effort of 1.5 man years per year

2012 Xenon Separation VMM family, CC EAL5+

2012 DornerWorks ARLX

DO-178 Level A packages, IEC 62304,
ISO 26262, MILS EAL, ARINC 653
Support for commercial and FOSS guest OSes

Virtuosity OA

OpenGroup FACE certified

Future Airborne Capability Environment (FACE™) defines the software computing environment and interfaces designed to support the development of **portable components across the general-purpose, safety, and security profiles**. FACE uses **industry standards** for distributed communications, programming languages, graphics, operating systems, and other areas as appropriate.

2012

Xenon Separation VMM family, CC EAL5+

2012

DornerWorks ARLX

DO-178 Level A packages, IEC 62304, ISO 26262, MILS EAL, ARINC 653

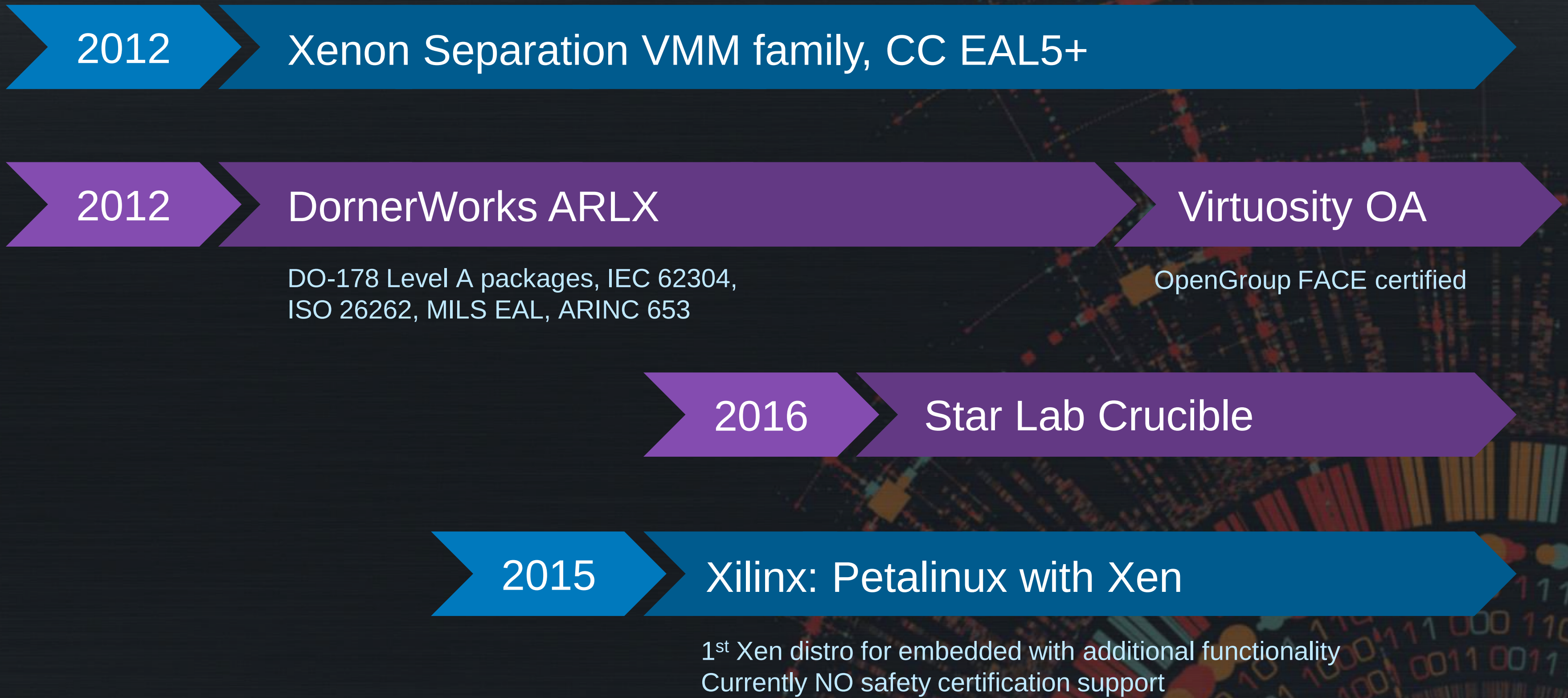
Virtuosity OA

OpenGroup FACE certified

2016

Star Lab Crucible

Secure embedded virtualization platform for security-critical operational environments, including aerospace & defense, industrial, transportation, and telecommunications



2012

Xenon Separation VMM family, CC EAL5+

2012

DornerWorks ARLX

DO-178 Level A packages, IEC 62304,
ISO 26262, MILS EAL, ARINC 653

Virtuosity OA

OpenGroup FACE certified

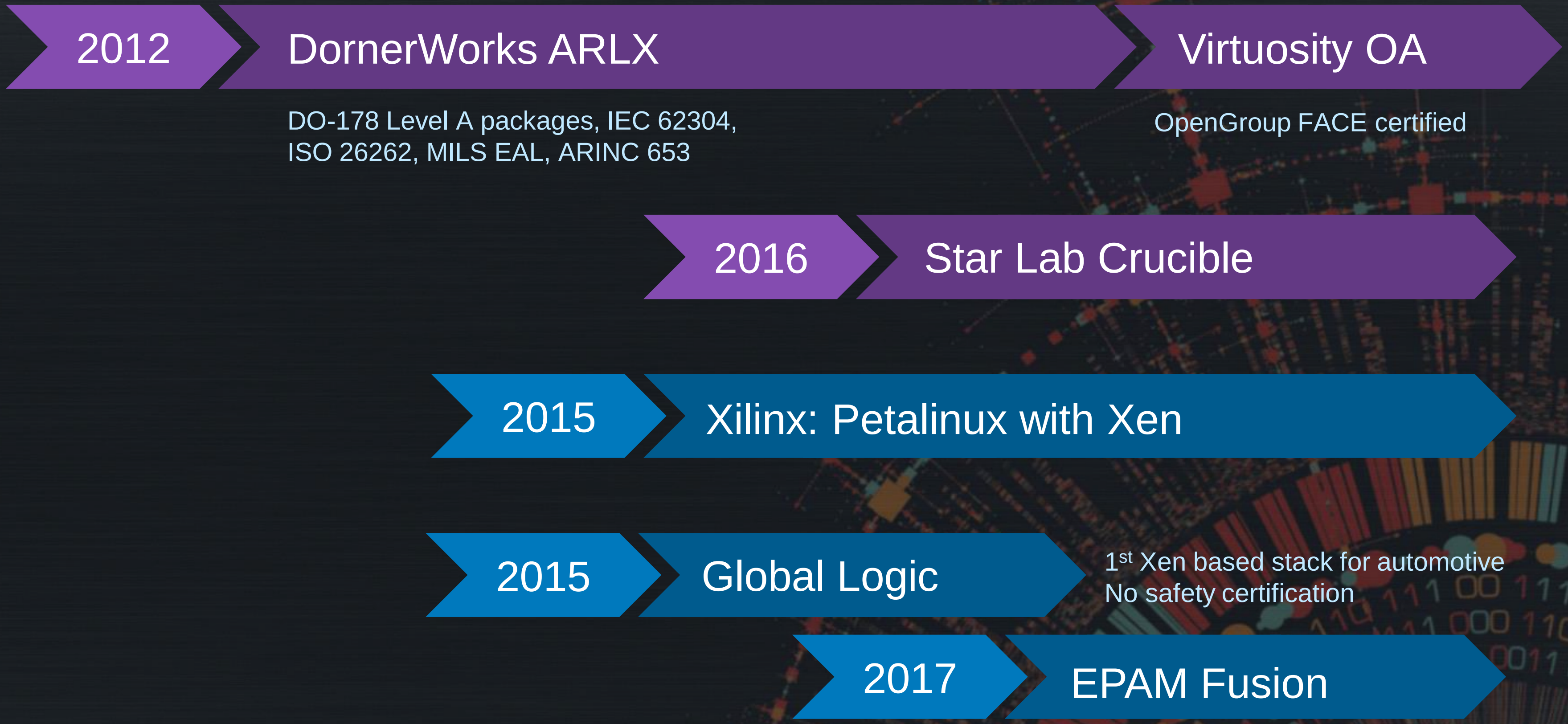
2016

Star Lab Crucible

2015

Xilinx: Petalinux with Xen

1st Xen distro for embedded with additional functionality
Currently NO safety certification support



2012 DornerWorks ARLX

DO-178 Level A packages, IEC 62304,
ISO 26262, MILS EAL, ARINC 653

Virtuosity OA

OpenGroup FACE certified

2016 Star Lab Crucible

2015 Xilinx: Petalinux with Xen

2015 Global Logic

1st Xen based stack for automotive
No safety certification

2017 EPAM Fusion

2nd generation Xen based stack for automotive. No safety certification, but working with community and industry on progressing safety

Summary

2016:
EPAM and Renesas funded a study by HORIBA MIRA to assess whether it is possible to safety certify a subset of the Xen Project

Answer: possible

From 2015 – today:
Close functional gaps, real-time capability, reducing code-size and create reference implementations (Arm, EPAM, XILINX)

Answer: suitable platform for some use-cases
Number of gaps to be a general purpose platform still worked on

All is open source, but not all is upstreamed in Xen

The impact on the Xen Project

Features specific to Embedded

Schedulers: ARINC, RTDS, Null and other real-time support

Laid the foundation for embedded use-cases and use of Xen as a partitioning HV

Low latency and real-time support

A minimal Xen on Arm Configuration

< 50 KSLOC of code for a specific HW environment

PV drivers (and in future virtio drivers) and GPU mediation for rich IO

Available in various upstreams

OP-TEE virtualization support

Both in Xen and in OP-TEE

Dom0less Xen

For now: allows booting VM’s without interaction with Dom0, but Dom0 still exists

2020: an architecture without a Dom0 and/or an RTOS as Dom0

Features specific to Embedded

Schedulers: ARINC, RTDS, Null and other real-time support

Laid the foundation for embedded use cases and use of Xen as a partitioning HV

Low

A r

< 5

PV O

Ava

OP

Bot

Do

For now: allows booting VM’s without interaction with Dom0, but Dom0 still exists

2020: an architecture without a Dom0 and/or an RTOS as Dom0

Key Point:

Xen on Arm, turned out to be a great open source hypervisor for embedded and mixed-criticality use-cases

Despite having been designed for servers!

Safety Certification
The Final Frontier

Attempts to solve this problem

FreeRTOS / SafeRTOS

FreeRTOS-compatible alternatives from Wittenstein
SafeRTOS: proprietary FreeRTOS-rewrite complying with IEC 61508

SIL2LinuxMP

Can Linux be Safety certified? Obstacles, tools and processes

LF Projects with an ambition to become ”easy to certify”

ACRN
AGL – Virtualization may make achieving key AGL UCs easier
ELISA Project – Develop tools and processes
Xen Project
Zephyr

Each with different history, cultures and problems that have to be overcome

FOSS SW and Functional Safety

Can FOSS SW be used for Functional Safety?
Yes, but there are many barriers

Requires major changes to the software

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

Certification Costs: Example DO-178

Level	Requirements	Application	Cost with Experience
DAL E	The software must exist	**Infotainment** Failure is a minor inconvenience	0.11 hour / SLOC
DAL D	High-Level Docs/Tests	**Instruments** Failure can be mitigated by operator	0.13 hour / SLOC
DAL C	Low-Level Docs/Unit Tests, Statement Coverage, and Code/Data Coupling Analysis		0.20 hour / SLOC
DAL B	Branch Coverage	**Engine Control** Failure could kill someone without warning	0.40 hour / SLOC
DAL A	Source to Object Analysis and MC/DC Coverage		0.67 hour / SLOC

Credit/Source: Dornerworks / XPDS14 - Xen and the Art of Certification.pdf

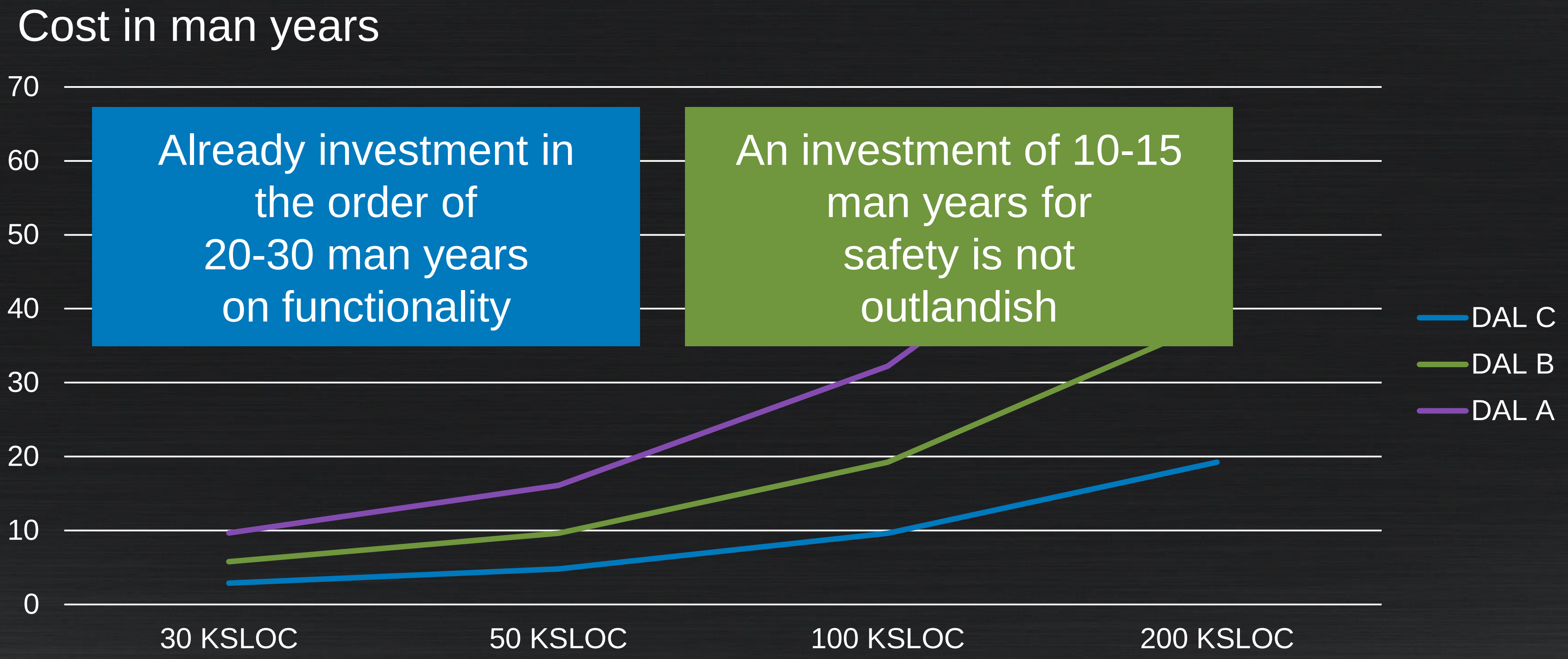
Certification Costs: Example DO-178

Level	Requirements	Application	Cost with Experience
DAL E	The software must exist	**Infotainment** Failure is a minor inconvenience	0.11 hour / SLOC
DAL D	High-Level Docs/Tests	**Instruments** Failure can be mitigated by operator	0.13 hour / SLOC
DAL C	Low-Level Docs/Unit Tests, Statement Coverage, and Code/Data Coupling Analysis		0.20 hour / SLOC
DAL B	Branch Coverage	**Engine Control** Failure could kill someone without warning	0.40 hour / SLOC
DAL A	Source to Object Analysis and MC/DC Coverage		0.67 hour / SLOC

3-4 times as much without experience

Credit/Source: Dornerworks / XPDS14 - Xen and the Art of Certification.pdf

Certification Costs: Example DO-178



Already investment in the order of 20-30 man years on functionality

An investment of 10-15 man years for safety is not outlandish

Xen Project’s starting point

Examples of Xen based embedded products

With some support for safety standards in proprietary spin-offs

Expertise in ecosystem that covers Xen and Safety

Primarily for hire: too small to fund speculatively

Reference implementations with safety in mind

EPAM Stack (automotive), XILINX Stack

Another similar effort in progress elsewhere (generic safety case)

Some limited adoption in niche use-cases today

In a non-safety context

In safety contexts where safety can be isolated in production/in progress

Where we want to be

Want to be in a position where upstream and vendors interested in safety certification collaborate with the goal of making Xen more cheaply **safety certifiable**

With buy-in and support from multiple vendors

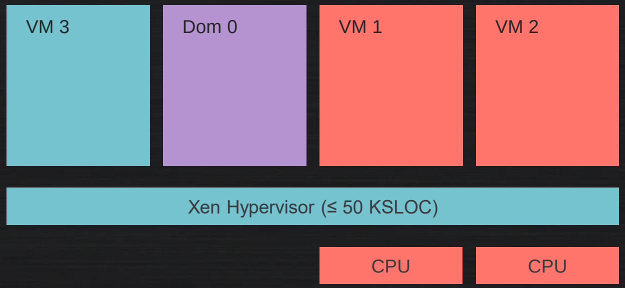
Don’t want to be at the bleeding edge of this, but just behind

Such that we can benefit from ELISA and other projects such as Zephyr

Safety Certification
The beginning of the journey

Mixed Criticality case

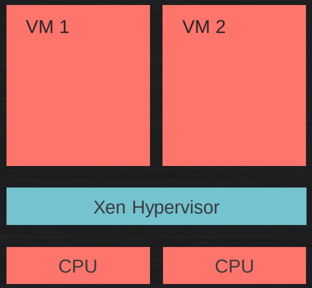
Dom0less VMs (today)



Dom0less VMs loaded by uBoot and booted by Xen (not Dom0), pinned to a CPU via the Null scheduler and I/O handled by device assignment

Dom0 completes boot after VM 1 and VM 2. Static set-up

True Dom0less (2019/20)



Ongoing work to fully implement true Dom0less for small systems

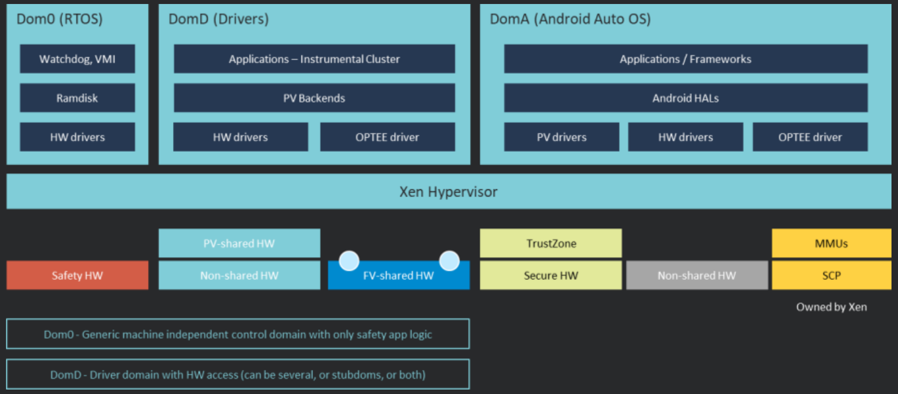
- Shared memory and interrupts for VM-to-VM communications
- PV frontends/backends drivers for Dom0-less VMs

Dom0less initial safety certification scope

Automotive Case

Mix Safety Digital Cockpit

In-Vehicle Computer



Dom0 (RTOS)
Watchdog, VMI
Ramdisk
HW drivers

DomD (Drivers)
Applications – Instrumental Cluster
PV Backends
HW drivers
OPTEE driver

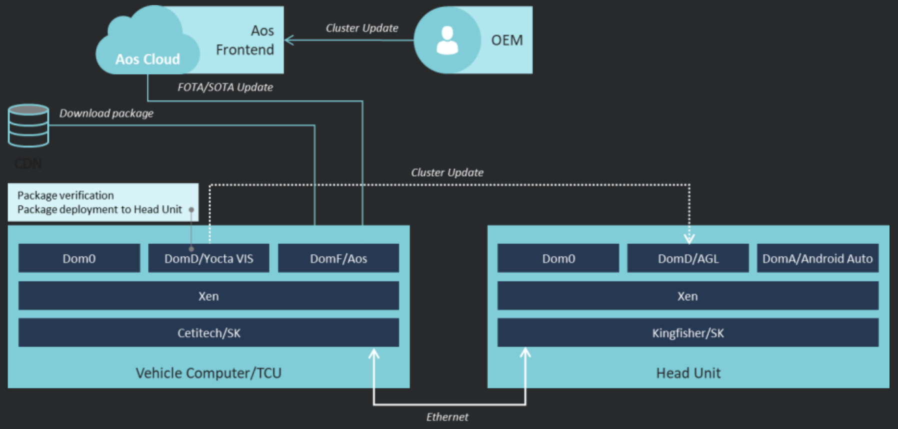
DomA (Android Auto OS)
Applications / Frameworks
Android HALs
PV drivers
HW drivers
OPTEE driver

Xen Hypervisor

Safety HW
PV-shared HW
Non-shared HW
FV-shared HW
TrustZone
Secure HW
Non-shared HW
MMUs
SCP
Owned by Xen

Dom0 - Generic machine independent control domain with only safety app logic

DomD - Driver domain with HW access (can be several, or stubdoms, or both)



Aos Cloud
Aos Frontend
Cluster Update
OEM
FOTA/SOTA Update
Download package
CDN
Cluster Update
Package verification
Package deployment to Head Unit

Dom0
DomD/Yocta VIS
DomF/Aos
Xen
Cetitech/SK
Vehicle Computer/TCU

Dom0
DomD/AGL
DomA/Android Auto
Xen
Kingfisher/SK
Head Unit

Ethernet

AUTOMOTIVE GRADE LINUX

Xen Project

androidauto

Community Challenges: MISRA C

Picked MISRA C as an example, because …

it is representative of the type of community problems that you should expect if you look at safety certification

Coding Standards: MISRA C

Required by most safety standards

10 Mandatory, 111 Required and 38 Advisory rules
Required rules depend on certification level can be deviated from
Justifications of deviations would have to be signed off by an assessor

Partnership with Perforce: access to QA Verify providing selected community members to results on Xen snapshots

Goal: Experiment and Learn

Picked hardest and controversial rules to see what would happen!

We did not expect to succeed !

We got stuck early on

MISRA C spec is proprietary

Rule text cannot be copied into a posted patch series ➜
lack of clarity, lack of rationale: leading to unnecessary debate

CI set-up does not allow upfront verification of fixes:

Primarily a consequence of what we were offered for free
Either: commit without knowing a fix worked
Or: The developer would have to buy the tool

Interactions w compilers, HW, assembly code problematic

Ended up with 11 iterations and man weeks of review effort

Bike shedding and strong opinions

Some rules will create a flame-war if there is a single argumentative maintainer

E.g. MISRA C:2012, 15.7
"if ... else if" constructs should end with "else" clause

```
if (x == 0) {
  doSomething();
} else if (x == 1) {
  doSomethingElse();
} else {
  error();
  /* or justification why no action is taken */
}
```

Deviations and Scalability

Possibility of MISRA C Deviations encourage arguments

Deviations: justification of a class or instance of non-compliance
Deviation Permits: previously approved deviations for a use-case

It’s all a bit like like “legal precedent” in common law legal systems:
an expert (assessor) is needed to advise the project on a case-by-case basis

Community Scalability

Code review process encourages too much discussion, if there is no up-front plan on how to approach a disruptive set of changes

Fix: A priori agreed strategy and plan on how to approach this

Safety Certification
Starting to plan …

Bring together Industry and Community

3 day workshop in March 2019 with 25 attendees – keep it small

Community Reps and Support

Project leadership team (except for 2)

Kate Stewart as observer/ advisor

THE LINUX FOUNDATION

Vendors with investment in Xen

arm

epam

RENESAS

XILINX

Safety Assessors

RESILTECH

exida

A HORIBA COMPANY MIRA

TÜV SÜD ISO 9001

Vendors with product interest

ADIT

STAR LAB SOFTWARE

PERFORCE

Objectives

Create a understanding between the community and industry

Terminology, Concepts, etc.
How safety certification works: look at different standards, routes, requirements
Explain assets and processes

Establish community “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

High Level Agreements

Split development model with an open and a closed part

Everything that is valuable to the wider community **ideally** in the open part,
e.g. documentation, **some** tests, 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 development workflow have to be kept minimal

There must be a benefit the community (including for common code)
Otherwise the community wont carry

There are long-term implications for the community

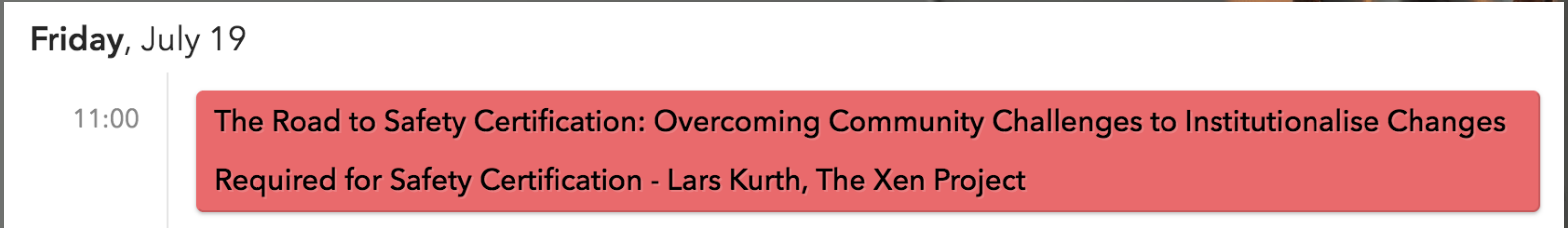
Make-up, scalability, decision making, conflicts – need to be managed
No new barriers for contributors can be introduced

Outcome: Is Xen Certifiable

Yes:

But assumes lightweight processes and automation in community
Similar to challenges using Agile in a safety context

What is next?



Friday, July 19

11:00 The Road to Safety Certification: Overcoming Community Challenges to Institutionalise Changes Required for Safety Certification - Lars Kurth, The Xen Project

Questions

Picture by Lars Kurth