

What are we trying to achieve?

- Identify and sign up Xen Project community members that are interested in ensuring that Xen is a candidate virtualization technology within AGL
- Contribute to the white-paper presenting a positive view of Xen which should help position Xen well within the AGL

Volunteers (signed up)

Rich Persaud (OpenXT) volunteered to coordinate Xen Project community

Lars Kurth (Citrix/Xen Project) will support Rich as much as he can

Stefano Stabellini can contribute to 2-3 Xen specific sections of the white-paper

Jonathan Kline (Star Lab) will contribute on Xen and safety certification

Artem Mygaiev (EPAM) has signed up to contribute to the white-paper prior to this discussion. **Within the Xen context and this group, the level of contribution is not yet clear.**

NEED: volunteers from companies which are more familiar with certification, Xen and the overall problem space

Immediate next actions:

- People on the preliminary discussion thread: Rich Persaud (OpenXT), Stefano Stabellini, Lars Kurth, Irby Thompson (StarLab), Christopher Clark (OpenXT), Alex Agizim (EPAM), Artem Mygaiev (EPAM)
People to include into the xen-devel thread: stewart.hildebrand@dornerworks.com, pattersonc@ainfosec.com, denys.balatsko@globallogic.com, yaroslav.bublykarvind@globallogic.com, arvind.murthy@globallogic.com, daniel.bernal@arm.com, jonathan.kline@starlab.io
- Feb 5th 1600 UTC, Xen community meeting on AGL white paper
- Feb 6th 1300 UTC, AGL EG-VIRT meeting (Lars and Stefano can't attend)

Summary of discussions amongst key stakeholders?

Lack of visibility of key concepts in [1]

Specifically, it is unclear on what information should be added under "AGL Virtualization approach", "AGL Virtualization architecture", and especially "Automotive hypervisor solutions".

Artem: Last time we spoke with Michele there was no clear understanding of what are the functional requirements from AGL to hypervisor, except generic understanding that some PV drivers and RT in some form must be supported.

Structure of white-paper may be aligned with other OSS or commercial hypervisors

We will need a list of Xen objections and responses, in addition to “pros” of Xen. What topics do we want to cover, to present Xen as a viable candidate within evaluation criteria that we co-define with other whitepaper contributors?

Which open-source and commercial guest-operating systems (AGL, RTLinux, OSADL, SIL2LinuxMP, Android, VxWorks, QNX) are being used by customers of AGL-based solutions? Which commercial, safety-certified, automotive hypervisors (Integrity, Wind River, QNX) are being considered by customers of AGL-based solutions?

Reference: <https://wiki.automotivelinux.org/bof-hypervisor>

Xen Project Selling points

Show the following key points about Xen (preliminary list). **We do need further input and expertise from Xen user companies that have experience in automotive and certification.**

1) Real time support: low latency, schedulers

We can show that Xen has a very low interrupt latency and comes with two real time schedulers support by default. At the same time, Xen can also run alongside VMs with a regular scheduler. KVM doesn't do real time well. Jailhouse doesn't do schedulers at all, but has low interrupt latency.

Irby: Only need RT scheduler if 2 VMs are sharing a core, otherwise use posted-interrupts / NULL scheduler to let RTOS schedule it's own tasks.

-- email from Stefano S to AGL mailing list on Jan 31 --

... people that are using [Xen] in environments with real time constraints usually configure it so that:

- a real time scheduler is used to schedule the workloads with real time constraints (for example, add sched=arinc653 on the xen command line)

- or, better, workloads with real time constraints have their vcpus statically assigned to pcpus

- there is no hypervisor paging: all the memory of the VM is allocated upfront (of course ballooning is still possible)

- there are no PV drivers, hardware devices are statically assigned to the VM with real time constraints (this requires an IOMMU)

- minimize interrupt latency: physical interrupts should be delivered to the pcpu running the vcpu that needs to receive the corresponding virtual interrupts. This is important. It is done by default by Xen on ARM.

This way, the scheduler overhead is null, the IO and memory overhead is down to stage2 translations in hardware, and the interrupt latency overhead is 5us (measured on a Xilinx Zynq MPSoC). This is usually within the workload's real time constraints.

2) Disaggregation/partitioning

Xen can be configured to partition the hardware into different VMs, similarly to what Jailhouse does. It relies on device assignment (no PV drivers).

Irby: “Separation kernel” is probably the most important case for why use a hypervisor in Automotive (i.e. - consolidate software functions onto fewer ECUs). Can we make a separation argument over KVM with its mixed hypervisor/kernel code? Focus on Xen’s dedicated device assignment to specific VMs.

3) PV drivers and driver domains

In addition to the above, Xen can export a device from an unprivileged guest to multiple other unprivileged guests using PV drivers. This is unique to Xen: neither Jailhouse nor KVM can do that.

Irby: Agreed that RT-compatible driver domains for multiplexed devices should be core part of Xen selling strategy.

4) Safety certifications

Although Xen is not certified, we believe it could be, due to the small code size. It’s far harder for KVM. It puts Xen in the same bucket with Jailhouse.

Irby: Where can we claim better features and hardware support over Jailhouse? What about simplified tooling to make deterministic (and certifiable) Xen-based configurations?

5) Very active community / ecosystem

This should be a strong counter-point to L4Re, and maybe seL4 and Jailhouse?

6) Open/not yet fully considered questions

1. Should we include case/studies and examples from vendors who have Xen Project based solutions in relevant market segments: candidates are Dornierworks, EPAM, GlobalLogic and StarLab. **If so, someone from these vendors would need to step up and contribute.**
2. Should we have a technical response to L4Re and SeL4 which are mentioned in a Linaro context in [4]. **seL4 is extremely limited, doesn’t support arm64 and has limited multicore support. L4Re is more advanced -- we need to find more detailed information about it.**
3. What can be usefully included from [5], [6] and [7]?
4. Concerns/questions raised by Irby Thompson (StarLab.io)

- How to prove freedom-from-interference on processors with shared L2/L3 cache, devices on a shared bus

Stefano: This is difficult and the results are likely to be platform specific. I think we need to run tests where a VM runs interference, and another VM runs a benchmark. We use the benchmarks results to prove that the interference is within acceptable parameters. Unfortunately I don’t have those numbers, but they would be very interesting to get.

Do we need to go into this level of details for the virtualization whitepaper?

- How to reduce the core Xen hypervisor codebase to something that can support full requirements and test case traceability (<20K SLOC)

Stefano: This should be possible with Kconfig. If not yet possible (I think we are slightly above 20K if I recall correctly), should be easy to make it so.

Business models

We need to consider a response to [1], section 4.2 which outlines some of the business/open source challenges on a high level. [4] touches on this from a more technical viewpoint also. It is not quite clear to me what AGL/industry thinking in these areas are.

Dornerworks appears to have gone down IEC 61508-3 Route 3s for ARLX already, specifically:

An argument can be made that taking an open source software element through safety certification after development is possible. A relatively small (<10K SLOC) software element could be snapshot (forked) and taken through an effort to reverse engineer requirements and validate the testing effort for certification. The result of such an effort would be a certified open source software element that would have to be managed as a new (forked) product.

It would be good to garner some of their views and experience. Note that the <10K SLOC “requirement” could be an issue.

1) Open/not yet fully considered questions

1. Concerns/questions raised by Irby Thompson (StarLab.io)
 - How to come up with either funding or a business model that supports the development of open-source certification artifacts?
Stefano: It is possible to formalize a consortium/organization to share the certifications costs among multiple members, but I wonder about the initial funding.
Lars: looking at the outline of [1] section 4.2 it appears to be understood that this is a question independent of the actual technology. Although [4] implies that this may be easier to achieve within the context of a more constrained open source model for L4Re and SeL4, but not for KVM, Jailhouse, Xen.

White paper timeline/plan

Copied from [1] for convenience. More details about upcoming meetings can be found in [3].

- 2018, January the 26th (Kick-off)
 - Dissemination of the call for participation
 - First Table of Contents (ToC) and plan presented
- February 6th (EG-VIRT webex meeting, 2CET webex to be set up)
 - Review ToC, discuss proposal for contributions and assign contributors to sections
 - Start editing
- February 9th (EG-VIRT webex meeting)
 - Fix contributors and table of contents
 - Continue editing
- February 16th (EG-VIRT webex meeting, webex to be set up)
 - First contributions ready, document review
 - Preparation of the content for AGL AMM and definition of next steps (e.g., identification of target events)
- AGL AMM February 20-21th

- Michele will present the white paper status
 - The target event for the release should be identified
- Release of the white paper (not more than 30 days after AMM)

References

[1] [AGL Virtualization EG-VIRT community white paper](#)

[2] [\[agl-discussions\] Call for Participation - Virtualization White Paper](#)

[3] <https://wiki.automotivelinux.org/eg-virt>

[4] [TSC Sponsored BoF: Can Linux and Automotive Functional Safety Mix ? Take 2: Towards an open source, industry acceptable high assurance OS](#)

[5] [PORTING OPERATING SYSTEMS TO RUN IN XEN VIRTUAL MACHINES Jarvis Roach DornerWorks](#)

[6]

<http://dornerworks.com/wp-content/uploads/2017/07/Partitioning-and-Virtualization-in-an-Embedded-Environment-DornerWorks.pdf>

[7] <http://events17.linuxfoundation.org/sites/events/files/slides/Xen%20Automotive.pdf>