Virtualization - Embedded Systems

Finn Artmann

OTH Regensburg, Faculty of Computer Science and Mathematics, Germany

Abstract
Virtualization solutions within embedded systems play a crucial role, offering benefits in varied industries including automotive, aerospace and IoT. The contribution of this work is an overview of the current state-of-the-art, identifying open questions and outlining new developments. The methodology involves a systematic review of recent literature, which is categorized based on content and keywords. Central findings from existing studies are summarized according to different virtualization approaches and enhanced with new insights from the reviewed papers, allowing for future exploration of presented research directions.

1. Introduction

Virtualization has become a crucial component in embedded systems, even in safety-critical industries [LLC23] like automotive [CDSM22], aerospace [SAVV*22] and IoT [LT22]. It helps to reduce size, weight, power consumption and costs [ILK*23], and is used to consolidate multiple software systems on the same system-on-a-chip (SoC) [CDSM22]. However, as embedded systems become more complex, they require robust isolation properties [SWL*22, CCDSR22]. Despite the widespread adoption of virtualization in various industries, only a few of the reviewed papers ([LLC23, CCDSR22]) provide a general overview of the current state-of-the-art in virtualization for embedded systems. This work contributes an updated overview of virtualization in embedded systems. It identifies open questions and new developments in the research area, serving as a foundation for further research and a starting point for interested students.

2. Paper Selection Criteria

To refine the paper selection, a query mandates the inclusion of *('All Metadata': embedded) AND ('All Metadata': virtualization)*, excluding those with *('Document Title': network)*. For retrieval of scientific and engineering literature, the databases *IEEE Xplore* and *Springer Professional* with institutional access for IEEE Xplore and the “Professional Book Archive Wirtschaft + Technik“ license for Springer Professional are utilized. Consequently, only papers classified as open access or available through the aforementioned subscriptions are considered.

To ensure contemporary relevance, the search is limited to the last two years (2022-2023). The Springer Professional database returns 12 results, none of which are deemed relevant to the subject, while the IEEE Xplore database produces 43 results, with 28 meeting the relevance criteria. Additionally, information is gathered from the primary sources of the reviewed papers. Relevance in this context is ascribed to papers addressing virtualization techniques applied to embedded systems or presenting an overview within this domain. Evaluation follows a sequential process, initially considering the title, followed by the abstract, then introduction and conclusion, and finally the full text examination. Acknowledging time and resource constraints, this work does not claim to be exhaustive.

3. Literature Analysis

To extract relevant information from the selected papers, the following sub-questions are formulated. These are intended to provide a guideline for the literature analysis.

- What are the current industry use cases for virtualization in the embedded domain?
- What are the current approaches to virtualization in the embedded domain?
- Which issues or open question exist within current approaches?
- Are there any new developments which try to address these issues?

Further, in order to get a quantitative overview of the papers, they are categorized according to their content and provided keywords. It has to be noted that some papers can fall into multiple categories. The results of this categorization are presented in table 1.

4. Industry Use Cases

In the following, use cases for virtualization in the industry mentioned in the reviewed papers are presented. In the automotive domain, the transformation of vehicles to have an increased connectivity to the outside world introduces new safety and security challenges, which virtualization aims to solve [KR23]. Additionally, consolidation of Electronic Control Units (ECUs) can be achieved by virtualization [GW22], with the overall goal of reducing costs

Category	Related papers
Survey	2
Container	2
Hypervisor	13
Separation kernel / Microkernel	4
Unikernel	2
Mixed criticality	9
Certification	2
Real-time	12
Simulation	3
Hardware support (incl. FPGA, RISC-V)	7
Energy/Power	4
Education	1
Security	2
Cellular devices	1
Automotive	5
Aerospace	3

Table 1: *Categorization of reviewed papers from the IEEE Xplore database.*

[AHFK22]. Trying to consolidate components by leveraging virtualization is not limited to automotive systems, but is also used in other domains (e.g. cellular devices [PPL22]). Virtualization is also utilized to construct simulation platforms for embedded systems, which improve research efforts by reducing experimental cycles and resource investments [HCDD22] or allow exploration and optimization of functional safety and security [KR23].

5. Virtualization Approaches

5.1. General-purpose Hypervisors

While general-purpose hypervisors like KVM and Xen were mostly used in server virtualization [CCDSR22], they are also currently used in embedded systems when properly tuned [AF19, Bon15, XLL*15, The18]. Xen for example is widely used in Xilinx Zynq platforms [ALJ*22]. For Xen, efforts have been made to optimize the schelduling algorithms of the virtual CPUs as well as to improve the interrupt handling [XWLG11, XXL*14, JYY11, GCGV06, GCN*09, ALJ*22].

Müller et al. [MAK22] highlight a research gap in open-source virtualization solutions for embedded systems and high-performance computing units. Their study evaluates KVM's performance on an NVIDIA DRIVE AGX Xavier Developer Kit, revealing poor performance and significant overhead, limiting its practical usability in real-world scenarios. Analysis of introduced latencies by KVM [AF19] show, that it is suitable for use in some real-time scenarios, while other sources explain how requirements can not be met or only with significant effort [SWL*22, MTS*20, Bon15].

Limitations of KVM and Xen are that they depend on the Linux kernel and that verification for functional correctness is difficult, if not impossible [HMS*23]. Form a security perspective, due to the size of their code base, they also have an increased attack surface [HMS*23].

5.2. Separation Kernels and Microkernels

Separation kernel and microkernel solutions aim for a low-complexity hypervisor while providing a high level of isolation, fitting well for embedded systems due to factors like dependability, certification, testing requirements and board platform support [CCDSR22]. Recent hypervisors combine separation kernel and virtualization concepts, enabling isolation of virtual machines at different criticality levels on the same hardware platform [CCDSR22]. Virtualization solutions based on separation kernels are also referred to as partitioning hypervisors [CCDS22]. They statically partition all platform resources and assign each one exclusively to a single virtual machine [MTS*20].

Representatives include VxWorks MILS (as cited by [CCDSR22]), Xtratum [CRM*09], and Jailhouse [Sie]. L4 microkernel-based solutions like PikeOS [sys], NOVA [SK10] and seL4 [KEH*09], the latter being formally verified for implementation correctness, are also mentioned [CCDSR22, HMS*23]. Generally, the most advanced solutions regarding certification are proprietary [CCDSR22]. In a recent paper by Cinque et al. [CDSM22], the authors try to identify a direction for certifying the open-source hypervisor Jailhouse according to the ISO 26262 standard. They assess the hypervisor using fault injection testing and were able to find some possible criticalities, leading to the hypervisor malfunctioning.

Martins and Pinto [MP23] analyze the static partitioning hypervisors (SPHs) Jailhouse, Xen (Dom0-less), Bao and seL4 CAmkES VMM. They highlight challenges in enhancing them. These include addressing inter-core interference, optimizing interrupt injection paths, achieving a balance between the simplicity of monolithic SPHs and the flexibility of microkernels, minimizing critical VM boot time overhead and ensuring secure full I/O passthrough mechanisms.

Partitioning hypervisors like Jailhouse and Bao aim to solve performance and security concerns of traditional hypervisors, but rely on static partitioning, assuming no resource sharing among guests [SWL*22]. To address this, Shen et al. [SWL*22] propose *Shyper*, a hypervisor employing hierarchical resource isolation strategies.

5.3. Unikernels

Unikerels, single-purpose applications compiled with necessary libraries and a thin OS layer [OLC*22], are gaining popularity for embedded domains as lightweight virtualization solutions [CCDSR22]. Despite their potential advantages, industry adoption is slow [MLS*17], due to challenges in porting existing applications to unikernel models [MLS*17, OLC*22, KBL*21]. *HermiTux* [OLC*22] addresses this by offering system call level binary compatibility with Linux applications, enabling it to run them without additional porting effort. However, open questions remain, including the need for stronger isolation proofs for certification in this domain [CCDSR22]. While many of the reviewed papers thematize real-time and mixed-criticality systems (table 1), only one survey paper [CCDSR22] addresses unikernels in this context, which concludes more analysis regarding their feasibility in aforementioned domains is required.

5.4. Containers

Container-based solutions, falling into the category of Operating System level virtualization, offer a lighter form of virtualization without emulating physical hardware compared to hypervisor-based solutions [CCDSR22]. While gaining momentum in real-time systems, further analysis is needed, especially regarding certification and isolation testing tasks in industrial contexts [CCDSR22].

Industrial domains like automotive and avionics, typically employ embedded systems. Containers show promise here, but are not yet fully mature for real-time capabilities [SBAP20]. Challenges identified by Struhár et al. [SBAP20] include the need for tools in real-time container management, communication between containers addressing real-time and security requirements and miscellaneous issues, like safety and security analysis, performance tests and resource utilization.

Stahlbock et al. [SWK22] discuss potential concerns for time-sensitive embedded systems, particularly the startup time of containers. They propose a concept to reduce preparation time during container creation, but acknowledge limitations such as experiments not reflecting real-world scenarios, assumptions about overhead and a lack of defined upper limits for acceptable startup times.

5.5. Hardware supported virtualization

Arm TrustZone is a hardware-based security extension for ARM processors, which allows to run a RTOS and a GPOS as isolated guests in a secure and non-secure environment [CCDSR22, PP22]. Regarding hardware-driven innovations in the virtualization area, Cinque et al. [CCDSR22] name ARM TrustZone-based solutions as main example in the embedded domain. According to the authors many embedded systems providers (e.g. Xilinx) build their production on top of ARM CPUS, which is why these are gaining momentum today.

A *type 0* hypervisor, a term introduced by Jansen et al. [JKD*17], is a hypervisor which is implemented in hardware. *BlueVisor* [JWD*22] is a recently proposed hypervisor falling into this category, promising higher performance and more predictability compared to software-based solutions. However, other sources [CCDSR22] classify these type of initiatives as currently still immature.

RISC-V, an open-source alternative to traditional instruction set architectures (ISA), is already supported by some hypervisors [LLC23, SMP22, RHS*22]. Adoption of this ISA shows there are still missing virtualization features, which are partly already being addressed, while other critical components to virtualization (e.g. IOMMU) are still missing [SMP22].

6. Power Consumption and Security

As embedded systems are often employed in an environment with limited power supply [HMS*23], some reviewed papers try to optimize it through consolidation of components by utilizing virtualization [PPL22], reduction of wakeup delays for a hypervisor [GW22] or solutions which dynamically scale clock frequency [KMI22, JYM*23]. Komori et al. [KMI22] apply *Dynamic Voltage Frequency Scaling* (DVFS) virtualization algorithms on a system and suggest applying it to multicore processors in the future, while the other papers provide more specific solutions with less of a general perspective.

The aspect of security is addressed in some reviewed papers [SWL*22, WCG22, OLC*22], however only in few, security is the main focus including a simulation platform for safety and security exploration [KR23] and runtime security monitoring for embedded hypervisors [HMS*23].

7. Discussion and Conclusion

While general purpose hypervisors have been adapted to embedded use cases, a currently more fitting approach for embedded systems are separation- and microkernel solutions, as they are developed with the embedded domain in mind. Containers and unikernels show potential, but industry adoption faces challenges. Advances in separation kernels and unikernels address issues like resource sharing and porting efforts. Hardware isolation features like ARM TrustZone are already well established, while new developments like type 0 hypervisors are still in an early stage. Additionally, RISC-V is gaining momentum and is already supported by some hypervisors. Future research is required across several areas, with separation- and microkernel posing more specific questions and unikernels and containers requiring more general analysis. Due to the limited scope of this work, a comprehensive analysis of all topics can not be assumed, future works could focus on specific topics and provide a more detailed analysis. This paper systematically reviewed a selection of papers and provides a concise summary of virtualization in embedded systems, highlighting central topics, open questions and new developments for future exploration.

References

[AF19] Abeni L., Faggioli D.: An Experimental Analysis of the Xen and KVM Latencies. In *2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC)* (May 2019), pp. 18–26. ISSN: 2375-5261. URL: https://ieeexplore.ieee.org/document/8759339, doi:10.1109/ISORC.2019.00014. 2

[AHFK22] Askaripoor H., Hashemi Farzaneh M., Knoll A.: E/E Architecture Synthesis: Challenges and Technologies. *Electronics 11*, 4 (Jan. 2022), 518. URL: https://www.mdpi.com/2079-9292/11/4/518, doi:10.3390/electronics11040518. 2

[ALJ*22] Alonso S., Lázaro J., Jiménez J., Muguira L., Bidarte U.: The influence of virtualization on real-time systems' interrupts in embedded SoC platforms. In *2022 37th Conference on Design of Circuits and Integrated Circuits (DCIS)* (Nov. 2022), pp. 01–06. ISSN: 2640-5563. URL: https://ieeexplore.ieee.org/document/9970041, doi:10.1109/DCIS55711.2022.9970041. 2

[Bon15] Bonzini P.: Realtime kvm. URL: https://lwn.net/Articles/656807/. 2

[CCDS22] Cesarano C., Cotroneo D., De Simone L.: *Towards Assessing Isolation Properties in Partitioning Hypervisors*. Tech. rep., Sept. 2022. arXiv:2209.00405 [cs] type: article. URL: http://arxiv.org/abs/2209.00405, doi:10.48550/arXiv.2209.00405. 2

[CCDSR22] Cinque M., Cotroneo D., De Simone L., Rosiello S.: Virtualizing Mixed-Criticality Systems: A Survey on Industrial Trends and Issues. *Future Generation Computer Systems 129* (Apr. 2022), 315–330. arXiv:2112.06875 [cs]. URL: http://arxiv.org/abs/2112.06875, doi:10.1016/j.future.2021.12.002. 1, 2, 3

[CDSM22] Cinque M., De Simone L., Marchetta A.: Certify the Uncertified: Towards Assessment of Virtualization for Mixed-criticality in the Automotive Domain. In *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)* (June 2022), pp. 8–11. ISSN: 2325-6664. URL: https://ieeexplore.ieee.org/document/9833844, doi:10.1109/DSN-W54100.2022.00012. 1, 2

[CRM*09] Crespo A., Ripoll I., Masmano M., Arberet P., Jean-Jacques M.: XtratuM: An Open Source Hypervisor for TSP Embedded Systems in Aerospace. 31. 2

[GCGV06] Gupta D., Cherkasova L., Gardner R., Vahdat A.: Enforcing Performance Isolation Across Virtual Machines in Xen. In *Middleware 2006* (Berlin, Heidelberg, 2006), van Steen M., Henning M., (Eds.), Lecture Notes in Computer Science, Springer, pp. 342–362. doi:10.1007/11925071_18. 2

[GCN*09] Govindan S., Choi J., Nath A., Das A., Urgaonkar B., Sivasubramaniam A.: Xen and Co.: Communication-Aware CPU Management in Consolidated Xen-Based Hosting Platforms. *IEEE Trans. Computers 58* (Aug. 2009), 1111–1125. doi:10.1109/TC.2009.53. 2

[GW22] Golchin A., West R.: Jumpstart: Fast Critical Service Resumption for a Partitioning Hypervisor in Embedded Systems. In *2022 IEEE 28th Real-Time and Embedded Technology and Applications Symposium (RTAS)* (May 2022), pp. 55–67. ISSN: 2642-7346. URL: https://ieeexplore.ieee.org/document/9804614, doi:10.1109/RTAS54340.2022.00013. 1, 3

[HCDD22] He R., Chen J., Du C., Duan Y.: Research on embedded system simulation technology. In *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)* (2022), vol. 10, pp. 2597–2601. doi:10.1109/ITAIC54216.2022.9836882. 2

[HMS*23] Hui H., McLaughlin K., Siddiqui F., Sezer S., Tasdemir S. Y., Sonigara B.: A Runtime Security Monitoring Architecture for Embedded Hypervisors. In *2023 IEEE 36th International System-on-Chip Conference (SOCC)* (Sept. 2023), pp. 1–6. ISSN: 2164-1706. URL: https://ieeexplore.ieee.org/document/10256735, doi:10.1109/SOCC58585.2023.10256735. 2, 3

[ILK*23] Ibellaatti N., Lepape E., Kilic A., Akyel K., Chouayakh K., Ferrandi F., Barone C., Curzel S., Fiorito M., Gozzi G., Masmano M., Navarro A. R., Muñioz M., Gallego V. N., Cueva P. L., Letrillard J.-n., Wartel F.: HERMES: qualification of High pErformance pRogrammable Microprocessor and dEvelopment of Software ecosystem. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (Apr. 2023), pp. 1–5. ISSN: 1558-1101. URL: https://ieeexplore.ieee.org/document/10136921, doi:10.23919/DATE56975.2023.10136921. 1

[JKD*17] Janssen B., Korkmaz F., Derya H., Hübner M., Ferreira M. L., Ferreira J. C.: Towards a type 0 hypervisor for dynamic reconfigurable systems. In *2017 International Conference on ReConFigurable Computing and FPGAs (ReConFig)* (Dec. 2017), pp. 1–7. URL: https://ieeexplore.ieee.org/document/8279825, doi:10.1109/RECONFIG.2017.8279825. 3

[JWD*22] Jiang Z., Wei R., Dong P., Zhuang Y., Audsley N. C., Gray I.: BlueVisor: Time-Predictable Hardware Hypervisor for Many-Core Embedded Systems. *IEEE Transactions on Computers 71*, 9 (Sept. 2022), 2205–2218. URL: https://ieeexplore.ieee.org/document/9601212, doi:10.1109/TC.2021.3125226. 3

[JYM*23] Jiang Z., Yang K., Ma Y., Fisher N., Audsley N., Dong Z.: Towards Hard Real-Time and Energy-Efficient Virtualization for Many-Core Embedded Systems. *IEEE Transactions on Computers 72*, 1 (Jan. 2023), 111–126. URL: https://ieeexplore.ieee.org/document/9893331, doi:10.1109/TC.2022.3207115. 3

[JYY11] Jeong J.-W., Yoo S., Yoo C.: PARFAIT: A new scheduler framework supporting heterogeneous Xen-ARM schedulers. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)* (Jan. 2011), pp. 1192–1196. ISSN: 2331-9860. URL: https://ieeexplore.ieee.org/document/5766431, doi:10.1109/CCNC.2011.5766431. 2

[KBL*21] Kuenzer S., Bădoiu V.-A., Lefeuvre H., Santhanam S., Jung A., Gain G., Soldani C., Lupu C., Teodorescu S., Răducanu C., Banu C., Mathy L., Deaconescu R., Raiciu C., Huici F.: Unikraft: fast, specialized unikernels the easy way. In *Proceedings of the Sixteenth European Conference on Computer Systems* (New York, NY, USA, Apr. 2021), EuroSys '21, Association for Computing Machinery, pp. 376–394. URL: https://dl.acm.org/doi/10.1145/3447786.3456248, doi:10.1145/3447786.3456248. 2

[KEH*09] Klein G., Elphinstone K., Heiser G., Andronick J., Cock D., Derrin P., Elkaduwe D., Engelhardt K., Kolanski R., Norrish M., Sewell T., Tuch H., Winwood S.: seL4: formal verification of an OS kernel. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles* (Big Sky Montana USA, Oct. 2009), ACM, pp. 207–220. URL: https://dl.acm.org/doi/10.1145/1629575.1629596, doi:10.1145/1629575.1629596. 2

[KMI22] Komori T., Masuda Y., Ishihara T.: DVFS Virtualization for Energy Minimization of Mixed-Criticality Dual-OS Platforms. In *2022 IEEE 28th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)* (Aug. 2022), pp. 128–137. ISSN: 2325-1301. URL: https://ieeexplore.ieee.org/document/9904796, doi:10.1109/RTCSA55878.2022.00020. 3

[KR23] Kabir M. R., Ray S.: Virtualization for Automotive Safety and Security Exploration. In *2023 IEEE 16th Dallas Circuits and Systems Conference (DCAS)* (Apr. 2023), pp. 1–4. URL: https://ieeexplore.ieee.org/document/10130221, doi:10.1109/DCAS57389.2023.10130221. 1, 2, 3

[LLC23] Lozano S., Lugo T., Carretero J.: A Comprehensive Survey on the Use of Hypervisors in Safety-Critical Systems. *IEEE Access 11* (2023), 36244–36263. URL: https://ieeexplore.ieee.org/document/10092745, doi:10.1109/ACCESS.2023.3264825. 1, 3

[LT22] Li Y., Takada H.: iSotEE: A Hypervisor Middleware for IoT-Enabled Resource-Constrained Reliable Systems. *IEEE Access 10* (2022), 8566–8576. URL: https://ieeexplore.ieee.org/document/9684412, doi:10.1109/ACCESS.2022.3144044. 1

[MAK22] Müller T., Askaripoor H., Knoll A.: Performance Analysis of KVM Hypervisor Using a Self-Driving Developer Kit. In *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society* (Oct. 2022), pp. 1–7. ISSN: 2577-1647. URL: https://ieeexplore.ieee.org/document/9968908, doi:10.1109/IECON49645.2022.9968908. 2

[MLS*17] Manco F., Lupu C., Schmidt F., Mendes J., Kuenzer S., Sati S., Yasukata K., Raiciu C., Huici F.: My VM is Lighter (and Safer) than your Container. In *Proceedings of the 26th Symposium on Operating Systems Principles* (New York, NY, USA, Oct. 2017), SOSP '17, Association for Computing Machinery, pp. 218–233. URL: https://dl.acm.org/doi/10.1145/3132747.3132763, doi:10.1145/3132747.3132763. 2

[MP23] Martins J., Pinto S.: Shedding Light on Static Partitioning Hypervisors for Arm-based Mixed-Criticality Systems. In

2023 IEEE 29th Real-Time and Embedded Technology and Applications Symposium (RTAS) (May 2023), pp. 40–53. ISSN: 2642-7346. URL: https://ieeexplore.ieee.org/document/10155684, doi:10.1109/RTAS58335.2023.00011. 2

[MTS*20] Martins J., Tavares A., Solieri M., Bertogna M., Pinto S.: Bao: A Lightweight Static Partitioning Hypervisor for Modern Multi-Core Embedded Systems. doi:10.4230/OASIcs.NG-RES.2020.3. 2

[OLC*22] Olivier P., Lefeuvre H., Chiba D., Lankes S., Min C., Ravindran B.: A Syscall-Level Binary-Compatible Unikernel. *IEEE Transactions on Computers 71*, 9 (Sept. 2022), 2116–2127. URL: https://ieeexplore.ieee.org/document/9591434, doi:10.1109/TC.2021.3122896. 2, 3

[PP22] Pan R., Parmer G.: SBIs: Application Access to Safe, Baremetal Interrupt Latencies. In *2022 IEEE 28th Real-Time and Embedded Technology and Applications Symposium (RTAS)* (May 2022), pp. 82–94. ISSN: 2642-7346. URL: https://ieeexplore.ieee.org/document/9804674, doi:10.1109/RTAS54340.2022.00015. 3

[PPL22] Park B., Park C., Li G.: DRX mode implementation based on virtual machine. In *2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)* (Oct. 2022), pp. 1–4. URL: https://ieeexplore.ieee.org/document/9970959, doi:10.1109/ICECS202256217.2022.9970959. 2, 3

[RHS*22] Ramsauer R., Huber S., Schwarz K., Kiszka J., Mauerer W.: Static Hardware Partitioning on RISC-V: Shortcomings, Limitations, and Prospects. In *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)* (Oct. 2022), pp. 1–6. URL: https://ieeexplore.ieee.org/document/10152063, doi:10.1109/WF-IoT54382.2022.10152063. 3

[SAVV*22] Sanchez-Aguero V., Valera F., Vidal I., Nogales B., Cabezas J., Vidal C.: A virtualization approach to validate services and subsystems of a MALE UAS. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (May 2022), pp. 1–6. URL: https://ieeexplore.ieee.org/document/9798286, doi:10.1109/INFOCOMWKSHPS54753.2022.9798286. 1

[SBAP20] Struhár V., Behnam M., Ashjaei M., Papadopoulos A. V.: Real-time containers: A survey. In *Workshop on Fog Computing and the Internet of Things* (2020). URL: https://api.semanticscholar.org/CorpusID:216086064. 3

[Sie] Siemens AG: Jailhouse hypervisor source code. Accessed 11-12-2023. URL: https://github.com/siemens/jailhouse. 2

[SK10] Steinberg U., Kauer B.: Nova: A microhypervisor-based secure virtualization architecture - eurosys2010, 2010. Accessed 11-12-2023. URL: https://hypervisor.org/eurosys2010.pdf. 2

[SMP22] Sá B., Martins J., Pinto S.: A First Look at RISC-V Virtualization From an Embedded Systems Perspective. *IEEE Transactions on Computers 71*, 9 (Sept. 2022), 2177–2190. URL: https://ieeexplore.ieee.org/document/9606600, doi:10.1109/TC.2021.3124320. 3

[SWK22] Stahlbock L., Weber J., Köster F.: An Optimization Approach of Container Startup Times for Time-Sensitive Embedded Systems. In *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)* (Dec. 2022), pp. 2019–2026. URL: https://ieeexplore.ieee.org/document/10074694, doi:10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00300. 3

[SWL*22] Shen Y., Wang L., Liang Y., Li S., Jiang B.: Shyper: An embedded hypervisor applying hierarchical resource isolation strategies for mixed-criticality systems. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (Mar. 2022), pp. 1287–1292. ISSN: 1558-1101. URL: https://ieeexplore.ieee.org/document/9774664, doi:10.23919/DATE54114.2022.9774664. 1, 2, 3

[sys] PikeOS RTOS & Hypervisor. Accessed 11-12-2023. URL: https://www.sysgo.com/pikeos. 2

[The18] The Linux Foundation: The Automotive Grade Linux Software Defined Connected Car Architecture. URL: https://www.automotivelinux.org/wp-content/uploads/sites/4/2018/06/agl_software_defined_car_jun18.pdf. 2

[WCG22] Wulf C., Charaf N., Göhringer D.: Virtualization of Reconfigurable Mixed-Criticality Systems. In *2022 32nd International Conference on Field-Programmable Logic and Applications (FPL)* (Aug. 2022), pp. 54–60. ISSN: 1946-1488. URL: https://ieeexplore.ieee.org/document/10035124, doi:10.1109/FPL57034.2022.00020. 3

[XLL*15] Xi S., Li C., Lu C., Gill C. D., Xu M., Phan L. T., Lee I., Sokolsky O.: RT-Open Stack: CPU Resource Management for Real-Time Cloud Computing. In *2015 IEEE 8th International Conference on Cloud Computing* (June 2015), pp. 179–186. ISSN: 2159-6190. URL: https://ieeexplore.ieee.org/document/7214043, doi:10.1109/CLOUD.2015.33. 2

[XWLG11] Xi S., Wilson J., Lu C., Gill C.: RT-Xen: towards real-time hypervisor scheduling in xen. In *Proceedings of the ninth ACM international conference on Embedded software* (New York, NY, USA, Oct. 2011), EMSOFT '11, Association for Computing Machinery, pp. 39–48. URL: https://dl.acm.org/doi/10.1145/2038642.2038651, doi:10.1145/2038642.2038651. 2

[XXL*14] Xi S., Xu M., Lu C., Phan L. T. X., Gill C., Sokolsky O., Lee I.: Real-time multi-core virtual machine scheduling in Xen. In *2014 International Conference on Embedded Software (EMSOFT)* (Oct. 2014), pp. 1–10. URL: https://ieeexplore.ieee.org/document/6986113, doi:10.1145/2656045.2656061. 2