

# ACRN

## Consolidate Real-Time and HMI with ACRN Hypervisor

---

Jack Ren, Intel ACRN Team

---

# Table of Contents

PART 1: What is ACRN	.....	page 3
PART 2: Industrial Use Case	.....	page 6
PART 3: Architecture	.....	page 7
PART 4: Roadmap	.....	page 14
PART 5: Call for Participation	.....	page 15

# What is ACRN™

A flexible, open-source, lightweight hypervisor  
for IOT workload consolidation

**A Linux Foundation Project Launched in March 2018**



<https://projectacrn.org>

# Value Proposition



## Core Intrinsic Values



### Small Footprint

- Optimized for IOT class solutions
- Significantly smaller footprint than datacenter targeted hypervisors



### Heterogeneous Workloads Consolidation

- Real time & Non-Real time
- Functionally Safe & non-safe



### Open-source with Flexible Licensing

- BSD license enables proprietary Guest OS
- True Open source with a vibrant Community

## Other Key Values

### System Security

- Intel VT backed virtualization provides secure operating domains

### Secure Containers

- Intel VT backed KATA containers as virtual machines enables added security
- Kubernetes support for KATA enables ease of deployment & management

### Flexible Isolation Frameworks

- Traditional virtualization w/ Shared framework
- Safety implementation w/ Partitioned framework

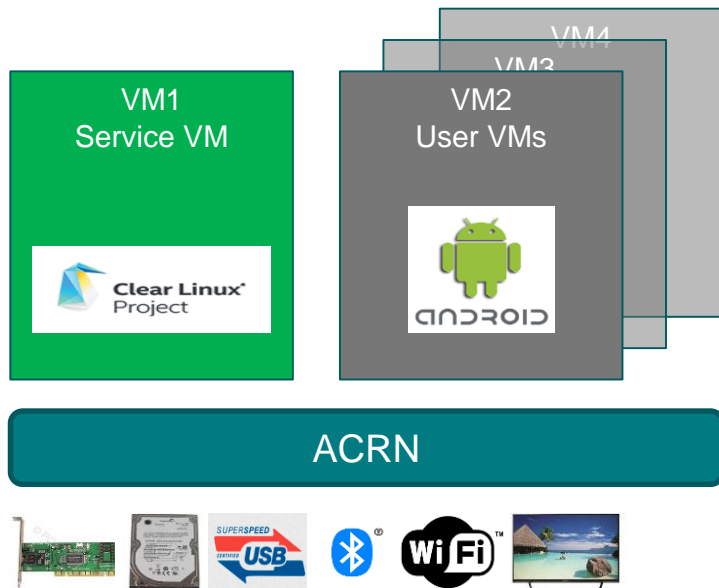
### Beyond-Compute sharing

- IO, Graphics, Media sharing capabilities

# ACRN 1.0



## Main Usage: Software Defined Cockpit



## Ready for Production

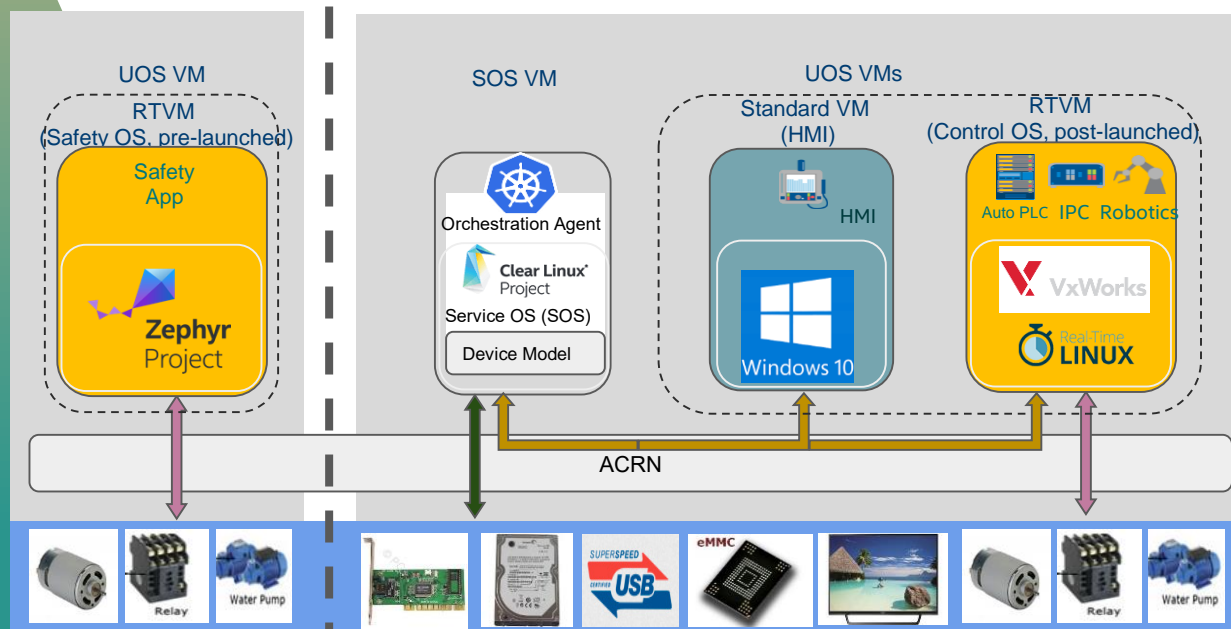
- **100% Feature Test Coverage**
- **High Stability**
- **Fast Boot and Performance KPI**
- **100% CTS Pass for Android Guest**

## Key Features

- **Safety and Security Isolation (Cluster + IVI)**
- **Extensive Sharing Capabilities**
  - Graphics, media, USB, audio, camera etc.
  - Advanced DMA/graphics buffer sharing
- **Multiple OS Support**
  - Clear Linux, Yocto, Ubuntu
  - Android, AliOS
- **MISRA-C Compliance**

**Released in May 2019 @[github.com/projectacrn/](https://github.com/projectacrn/)**

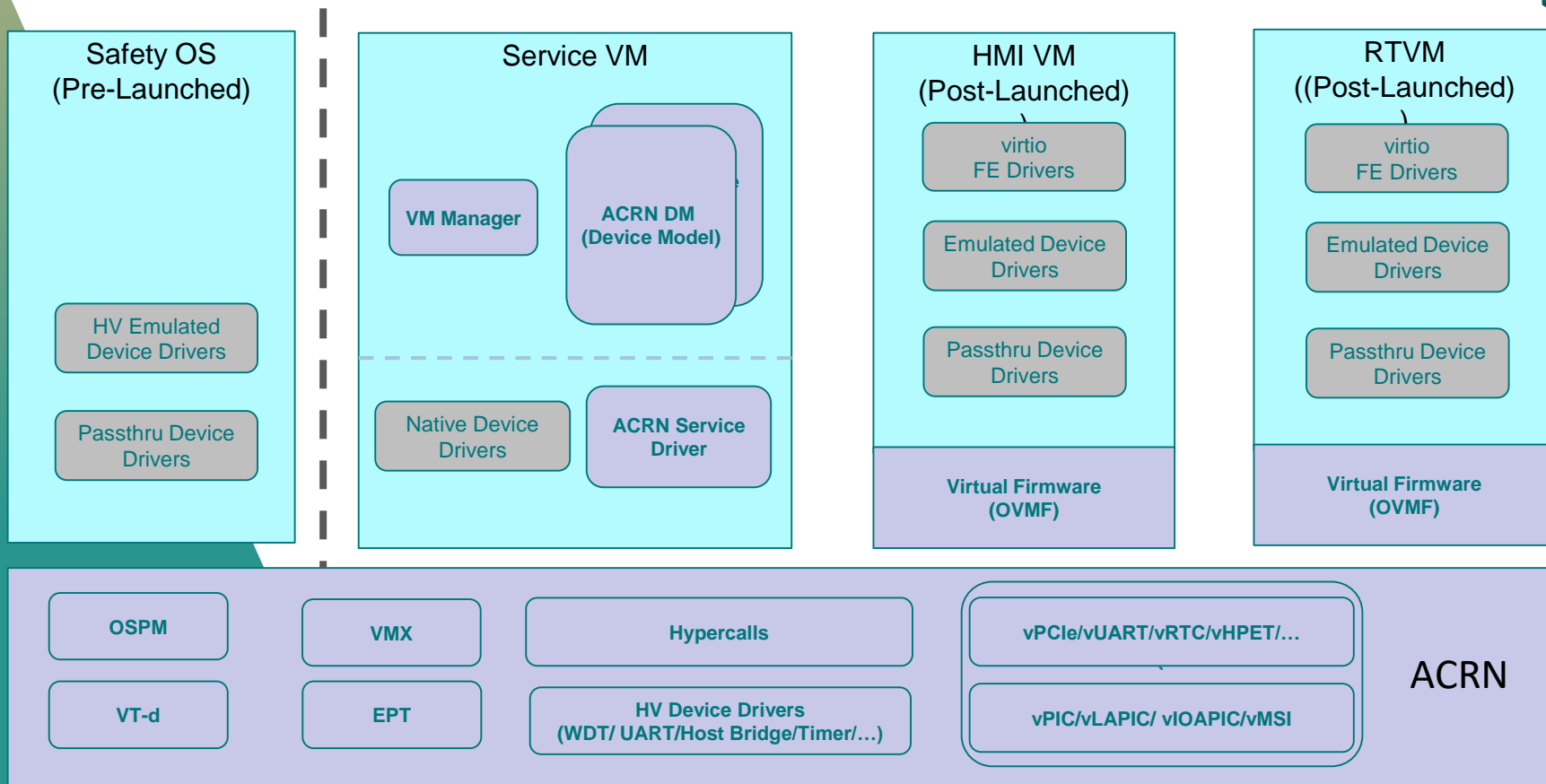
# Industrial: Safety + RT + HMI



## Key Challenges:

- ☐ **Mixed Criticality:**
  - Real-Time vs non Real-Time
  - Safety vs non-Safety
  - Isolation vs Sharing
- ☐ **Real-Time (Hard / Soft)**
  - GBE packet IO control loop < 12us
  - MSI interrupt latency < 4us
  - Cyclicttest jitter < 10us
- ☐ **HMI**
  - Window10
- ☐ **Functional Safety**
  - IEC 61508-3 (Industrial)
  - ISO 26262 (Automotive)

# ACRN 2.0 Architecture



# Industrial Usage: Device Mapping Table



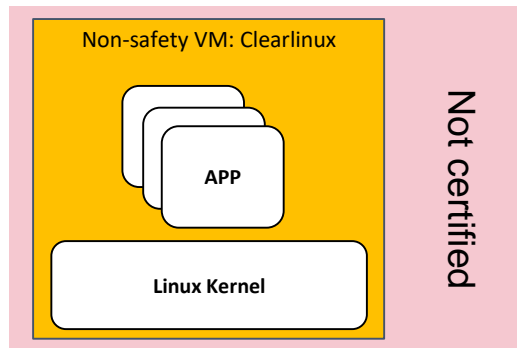
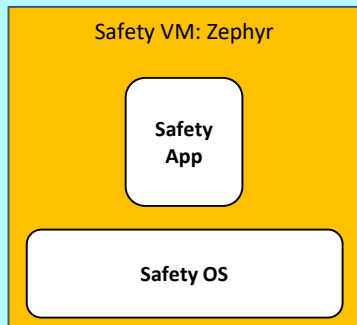
Devices	Soft RTVM (RT Linux)	Hard RTVM (VxWorks)	HMI (Windows/Linux)
RTC	Virtual	Virtual	Virtual
PCI	Virtual	Virtual	Virtual
UART	Passthru	Passthru	Virtual/Passthru
GBE Network	Virtual (PMD) /Passthru	Virtual (PMD) /Passthru	Virtual
TSN (i210)	Passthru	Passthru	N/A
Storage	Virtual (PMD) /Passthru	Virtual(PMD) /Passthru	Virtual
FPGA	Passthru	Passthru	N/A
GPU	N/A	N/A	Mediated Passthru
Audio	N/A	N/A	Passthru
USB	N/A	N/A	Virtual
Watchdog Timer (WDT)	Virtual	Passthru / Virtual	Virtual

# ACRN + Zephyr as Safety Domain

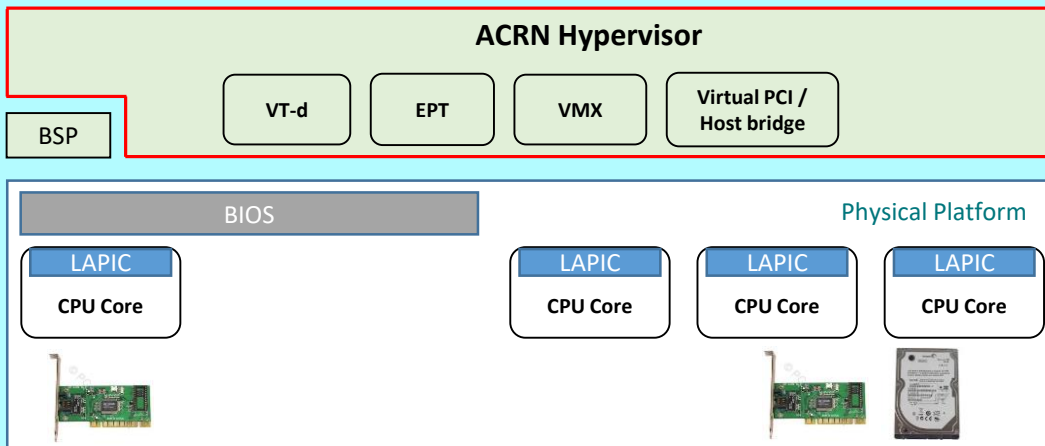
## Use Scenario



Certified by Intel and/or customers

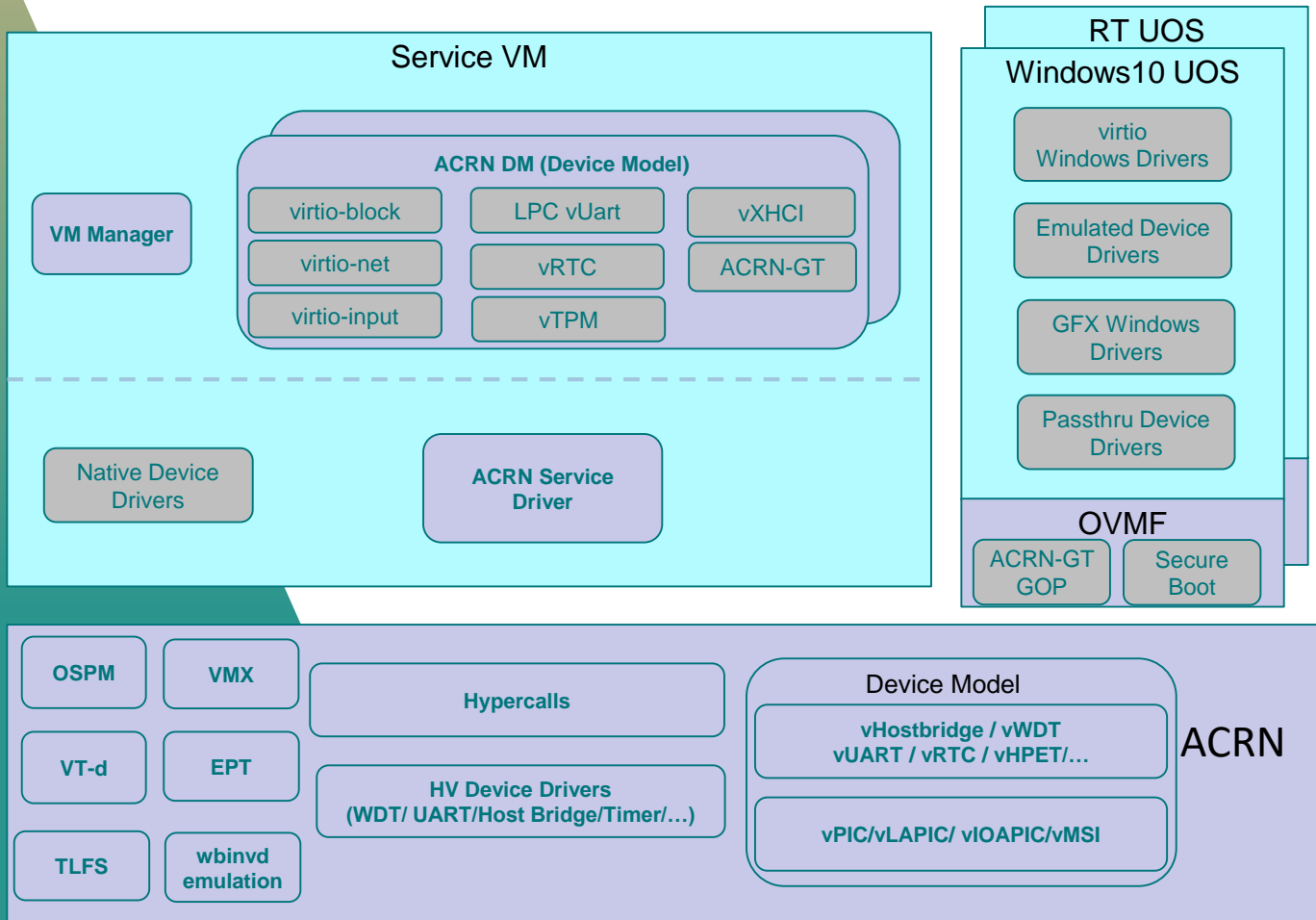


- ❑ 2 Partitions with mixed-criticality
- ❑ Static core & memory partitioning



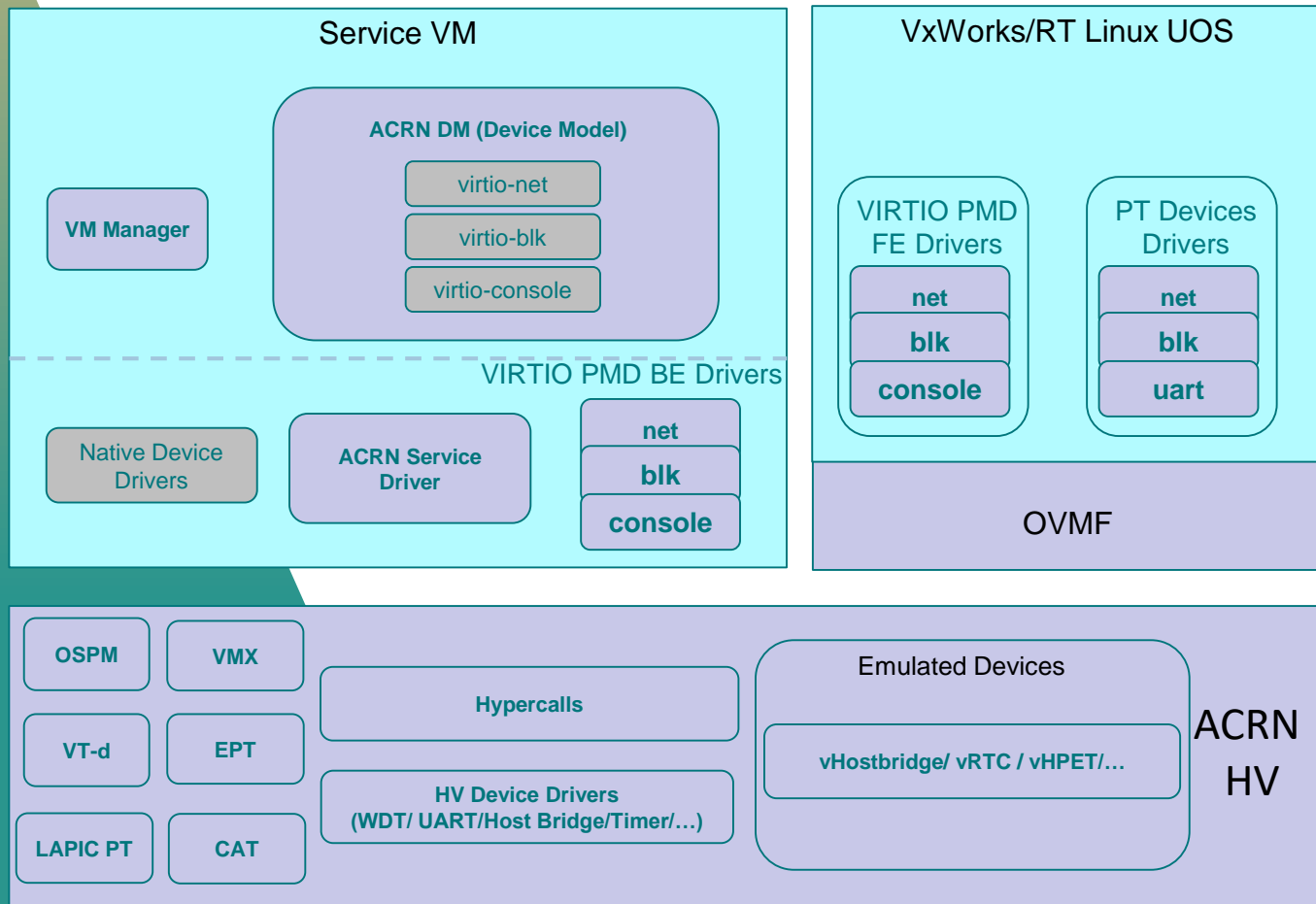
IEC 61508 & ISO 26262  
Certification Scope for  
ACRN

# Windows as HMI domain



- ❑ ACRN-GT GOP is added into OVMF to support windows early display and windows installation display.
- ❑ Support OVMF secure boot with vTPM for Windows secure boot chain.
- ❑ Support the Microsoft defined TLFS(Hyper-V Hypervisor Top-Level Functional Specification) minimum requirements and optional performance optimization requirements.
- ❑ Utilize Microsoft DISM tool to pre-install virtio-win drivers and gfx driver to the Windows install .iso file.
- ❑ Use GT-CLOS to prevent Windows from Cache interference

# VxWorks/RT Linux as Control domain



- ☐ Pass-through LAPIC (except ICR/XAPICD/LDR) to:
  - avoid VM-exit
- ☐ Enable CAT to:
  - isolate cache for RT VM
- ☐ Enable virtio BE/FE as PMD to:
  - avoid VM-exit
- ☐ Configure native BIOS to disable:
  - Hyper-threading
  - Speed Step
  - Speed Shift
  - C-state
  - GT RC6
  - GFX Lower Power Mode
  - Native ASPM
  - ...

# Configuration for Real Time Latency Evaluation



## Configuration:

- HW: Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz, 8G Memory, 1M L2 cache, 8M L3 cache
- Benchmark: cyclicttest (measure the scheduler jitter), running in Real-Time VM

## ACRN:

- Service OS VM: Linux kernel v4.14.68-rt42
- Real-Time VM: Preempt-RT Linux: 4.14.68-rt42, with 2GB memory

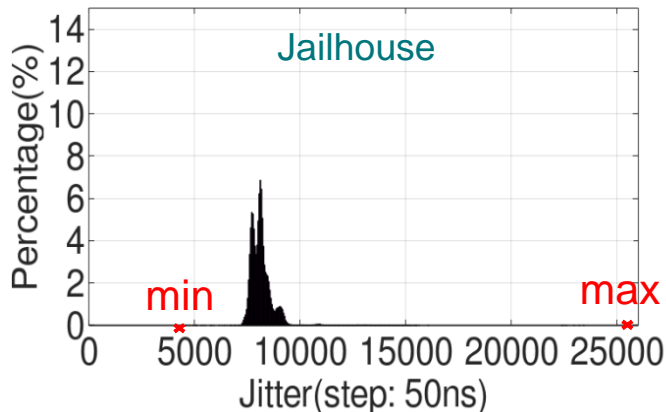
## Jailhouse:

- Root cell: Linux kernel v4.14.68-rt42
- Non-root cell: Preempt-RT Linux: Linux:4.14.71-rt44+, with 2GB memory

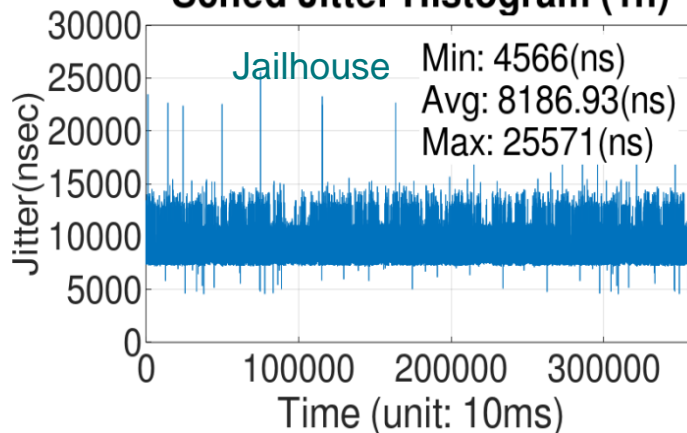
# cyclictest: ACRN vs Jailhouse



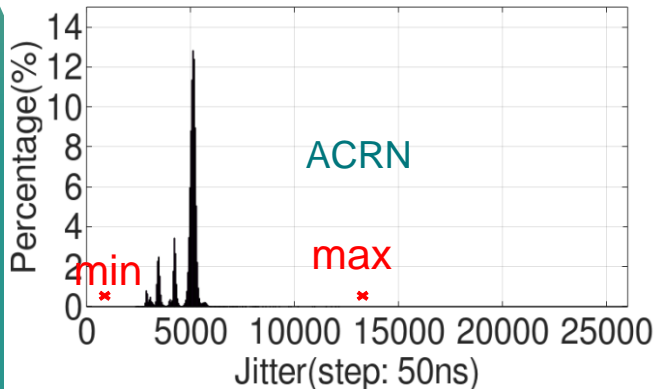
## Sched Jitter Distribution (1h)



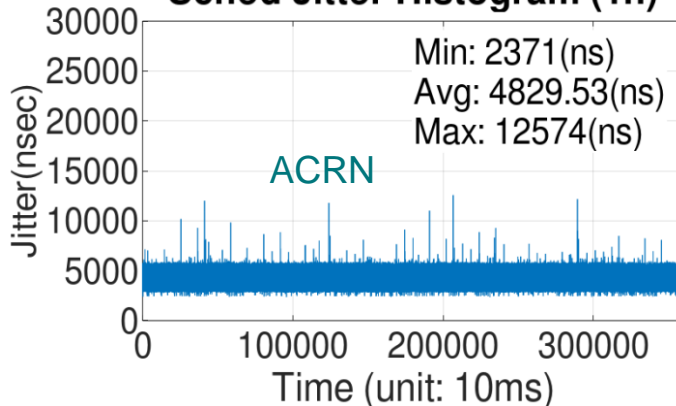
## Sched Jitter Histogram (1h)



## Sched Jitter Distribution (1h)



## Sched Jitter Histogram (1h)



# ACRN Open Source Roadmap in 2019



\*Feature and dates for reference only and subject to change without notices

Area	V1.0@Q1'19	Q2'19	Q3'19	Q4'19	2020
HW	<ul style="list-style-type: none"> <li>• APL NUC (UEFI)</li> <li>• KBL NUC (UEFI)</li> <li>• APL UP2 (SBL)</li> </ul>	<ul style="list-style-type: none"> <li>• APL NUC (UEFI)</li> <li>• KBL NUC (UEFI)</li> <li>• APL UP2 (SBL)</li> </ul>	<ul style="list-style-type: none"> <li>• APL NUC (UEFI)</li> <li>• KBL NUC (UEFI)</li> <li>• APL UP2 (SBL)</li> <li>• Denverton SoC</li> </ul>	<ul style="list-style-type: none"> <li>• APL NUC (UEFI)</li> <li>• KBL NUC (UEFI)</li> <li>• APL UP2 (SBL)</li> <li>• Denverton SoC</li> </ul>	
Hypervisor	<ul style="list-style-type: none"> <li>• Power Management (S3/S5)</li> <li>• ACRN partition mode</li> <li>• Local APIC passthrough</li> <li>• Real-Time VM support</li> </ul>	<ul style="list-style-type: none"> <li>• VxWorks as Guest</li> <li>• Zephyr as Guest</li> <li>• ACRN Real-Time baseline</li> <li>• ACRN Hybrid mode</li> <li>• OVMF for Clear Linux Guest support</li> <li>• IOMMU interrupt remapping</li> <li>• VM Configuration Unify</li> </ul>	<ul style="list-style-type: none"> <li>• Real-Time for Preempt-RT Linux</li> <li>• Real-Time for Pseudo Locking</li> <li>• Real-Time profiling tool</li> <li>• Real-Time Performance optimization</li> <li>• Kata Container support</li> <li>• OVMF GOP driver for GVT-g</li> <li>• Device Posted Interrupt(PI)</li> <li>• Multiple IOAPIC support</li> </ul>	<ul style="list-style-type: none"> <li>• Windows as guest</li> <li>• VxWorks as guest</li> <li>• Zephyr as Safety OS</li> <li>• CPU sharing</li> <li>• Docker support based on Kata Containers</li> </ul>	
I/O virtualization	<ul style="list-style-type: none"> <li>• GPIO virtualization</li> <li>• QoS – Support RunC</li> <li>• TPM2.0 Sharing (Security)</li> </ul>	<ul style="list-style-type: none"> <li>• SR-IOV for share mode</li> <li>• HPET Virtualization</li> <li>• Open vSwitch</li> <li>• I2C virtualization</li> </ul>	<ul style="list-style-type: none"> <li>• USB hub virtualization</li> </ul>	<ul style="list-style-type: none"> <li>• Kubernetes support based on Kata Containers</li> <li>• GVT-g Gen11 support</li> <li>• GVT-g for Windows as Guest</li> </ul>	

## Call to Action



<https://projectacrn.org>



### Join us!

If you support the ACRN project and feel that this is the right thing for the embedded ecosystem, join us in moving this project forward together as a community member.

We need code contributors, users, and project direction influencers!



### Contribute code!

Make a difference to the project by committing code, help us become a better project.

Project code merged in the past 6 months allows you to become a voting member of the Technical Steering Committee.



### All Contributions Matter

In open source projects a contribution can be anything which helps the project to accomplish its mission. Examples of Contributions beyond just code include:

Financial Assistance, Requirements Gathering, Documentation, Testing, Bug Reporting