ACRN

Project ACRN: A Comprehensive Overview

Last updated: December 13, 2021

Content

- Introduction
- Architecture
- Value Proposition
- Key Capabilities
- Scenarios

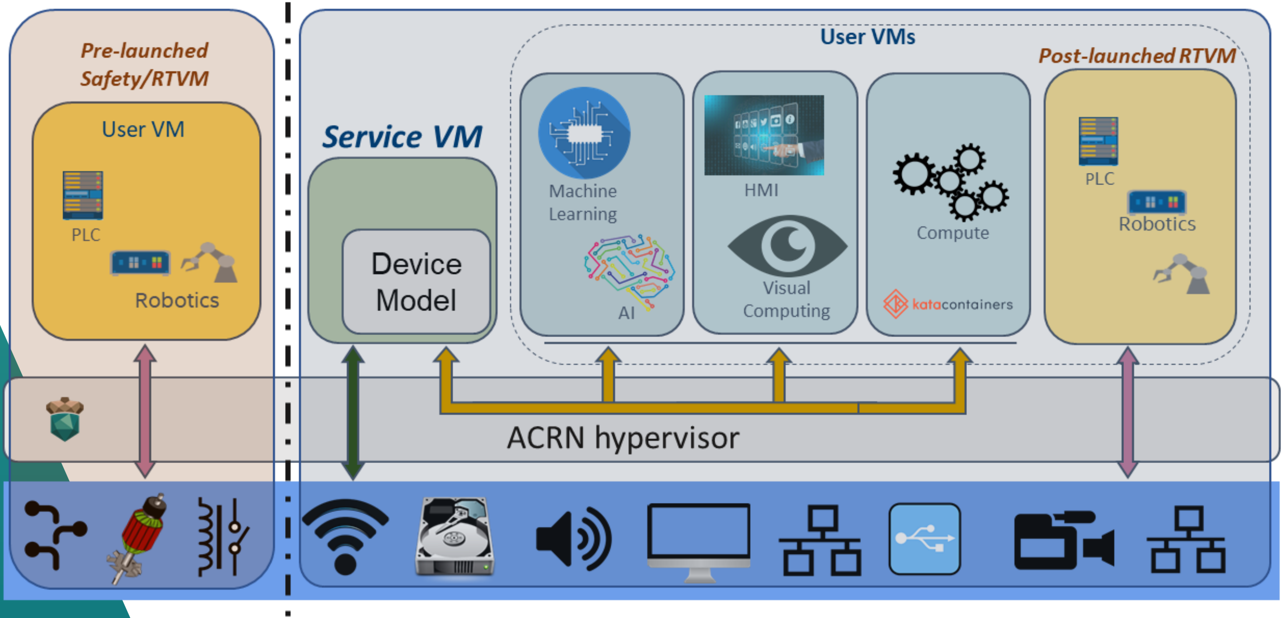
Introduction

ACRN™ is a flexible, lightweight reference hypervisor, built with real-time and safety-criticality in mind, optimized to streamline embedded development through an open source platform.

- A Linux Foundation Project Launched in March 2018
- Version 1.0 released in May 2019
- Version 2.0 released in June 2020
- Version 2.6 released in September 2021

https://projectacrn.org

Overall architecture



ACRN is a registered trademark of the Linux Foundation. *Other names and brands may be claimed as the property of others.

Value Proposition - ACRN

Small Footprint

- Optimized for resource-constrained devices
- Small codebase: less than 40,000 vs. >156,000 lines of code for datacenter-centric hypervisors

Security & Hard Real-time

- Heterogeneous Workloads Consolidation
- Supports also the most demanding workloads: security and real-time

Open-source with Flexible License

- Permissive BSD license enables proprietary Guest OS
- Business-friendly to adopt, modify and integrate
- True open source with a vibrant Community

ACRN reduces system deployment complexity, enables heterogeneous architectures, and provide TCO advantages

Key Capabilities

Hard Real-time

Support hard or soft RT VM
Optimized for RT, e.g. no VMExit*, cache isolation

Various Guest OSes Support

Linux*, Zephyr*, Android*, VxWorks*, Windows*…

Permissive Open Source License

Permissive BSD-3-clause license
Linux Foundation Affiliation

Rich I/O Mediation

Graphics, Audio, USB…
Industry standard Virtio BE/FE drivers

Secure Container

Kata Containers enabled for starting isolated
and secure containers

System Manageability

Flexible VM lifecycle Management
Virtualization API supported (libvirt)

Flexible Architecture

Shared, Partitioned and
Hybrid (mix of partitioned & shared) mode

Security & Isolation

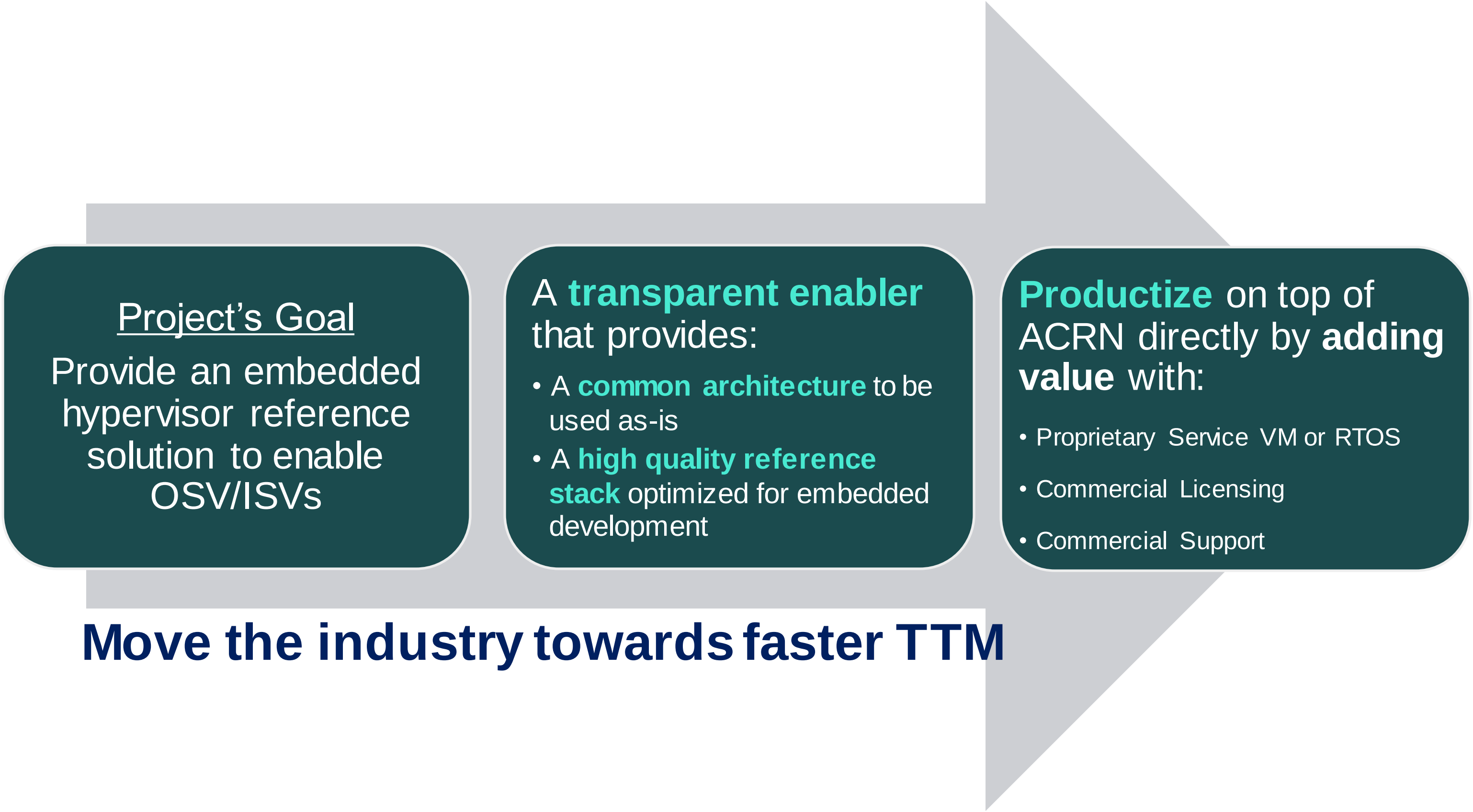
Full isolation for mixed criticality workloads
Intel VT backed isolation
Secure boot

Ease of use

ACRN configuration tool
Rich documentation
Multiple-channel community support

*Other names and brands maybe claimed as the property of others

ACRN™ & OSV/ISV Vendors



Project's Goal

Provide an embedded hypervisor reference solution to enable OSV/ISVs

A **transparent enabler** that provides:

- A **common architecture** to be used as-is
- A **high quality reference stack** optimized for embedded development

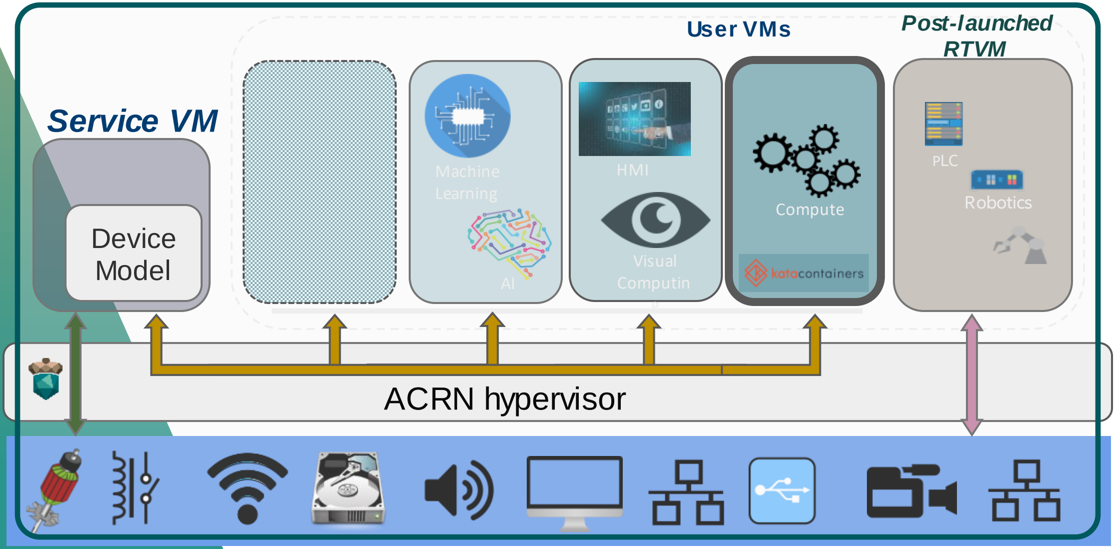
Productize on top of ACRN directly by **adding value** with:

- Proprietary Service VM or RTOS
- Commercial Licensing
- Commercial Support

Move the industry towards faster TTM

Shared Mode

Previously known as Industry



Key Challenges:

❑ **Mixed Workloads:**

- Real-Time vs non Real-Time
- Isolation vs Sharing

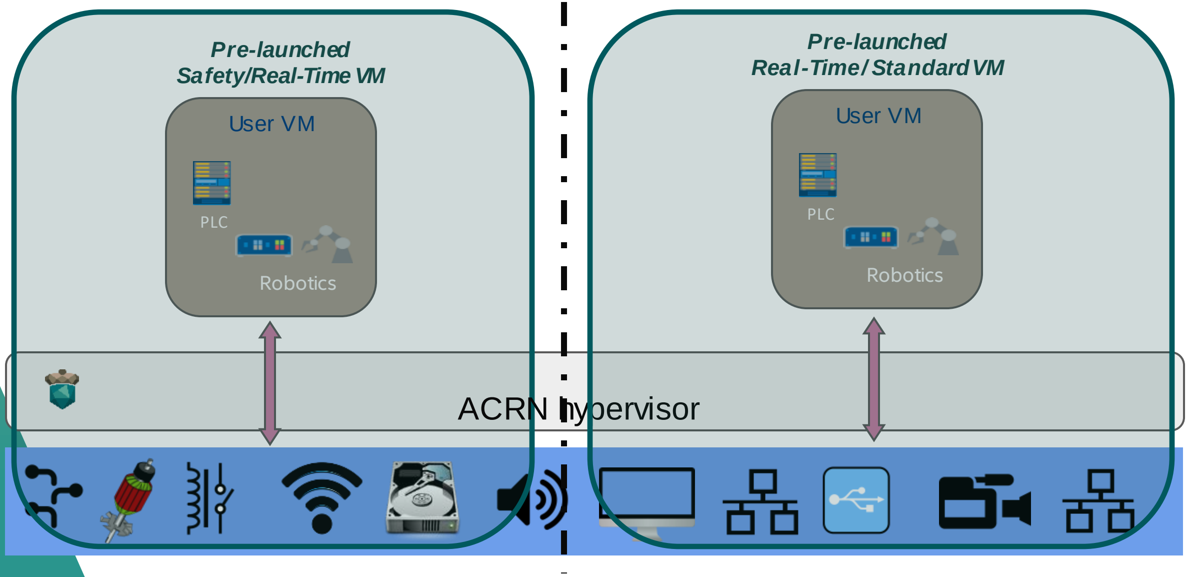
❑ **Real-Time (Hard / Soft)**

- GBE packet IO control loop < 12us
- MSI interrupt latency < 4us
- Cyclictest jitter < 10us

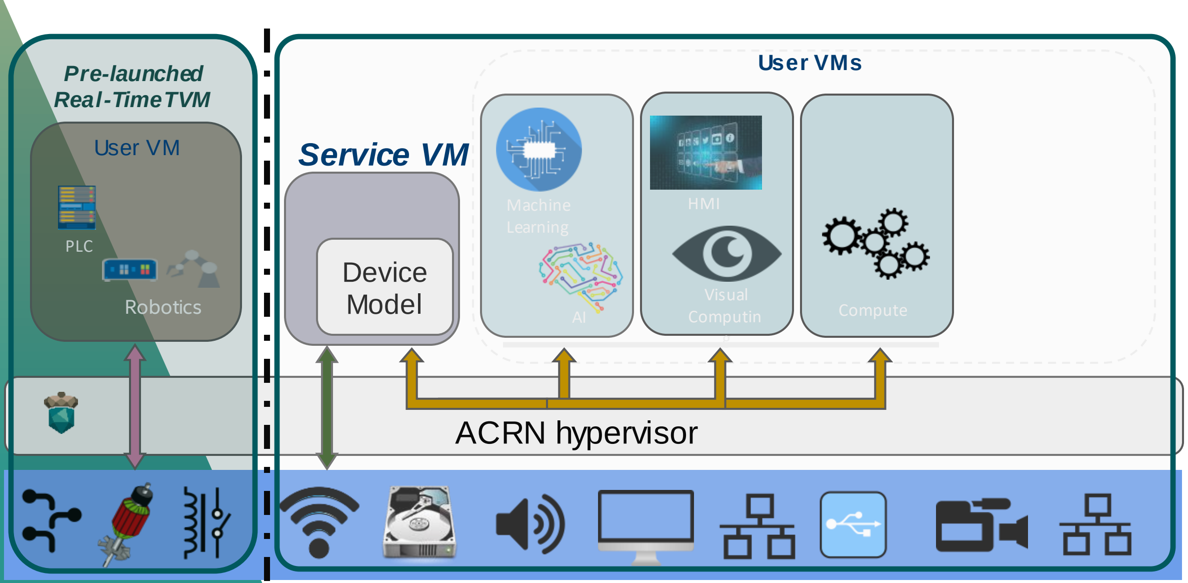
❑ **HMI**

- Windows 10

Partitioned Mode



Hybrid and Hybrid Real-Time



Key Challenges:

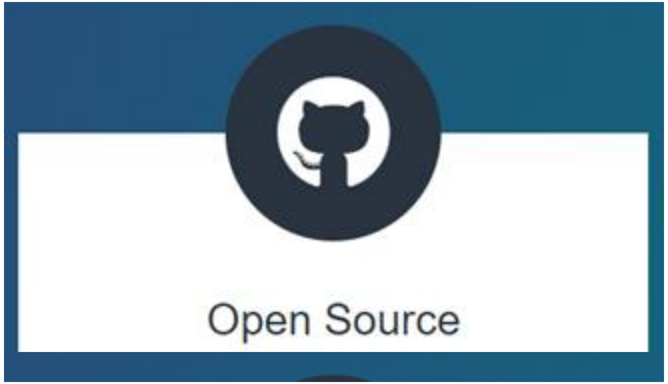
❑ **Mixed Workloads:**

- Real-Time and Standard workloads
- Strict Isolation and Sharing

❑ **Real-Time (Hard / Soft)**

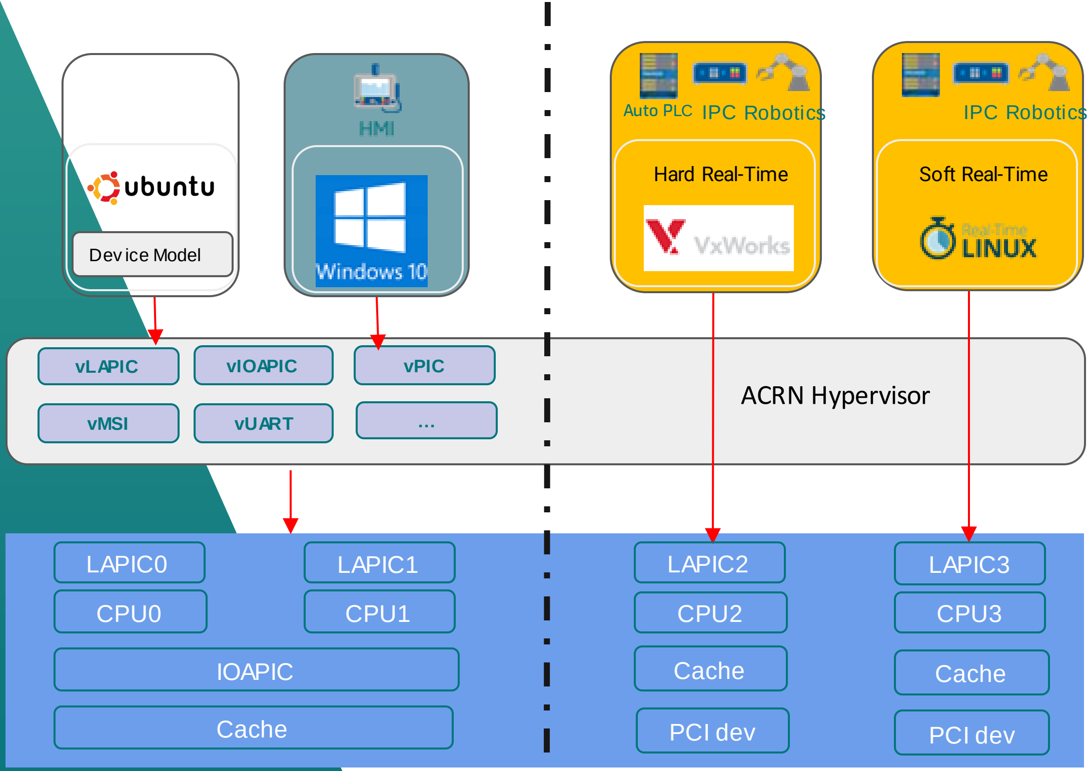
- GBE packet IO control loop < 12us
- MSI interrupt latency < 4us
- Cyclictest jitter < 10us

Open Source with Flexible Licensing



- Scalable support
- Significant R&D and development cost savings
- Transparent open source development model
- Collaborative SW development with industry leaders
- Permissive BSD licensing

Real Time



❑ Support hard or soft Real-Time VMs (RTVMs)

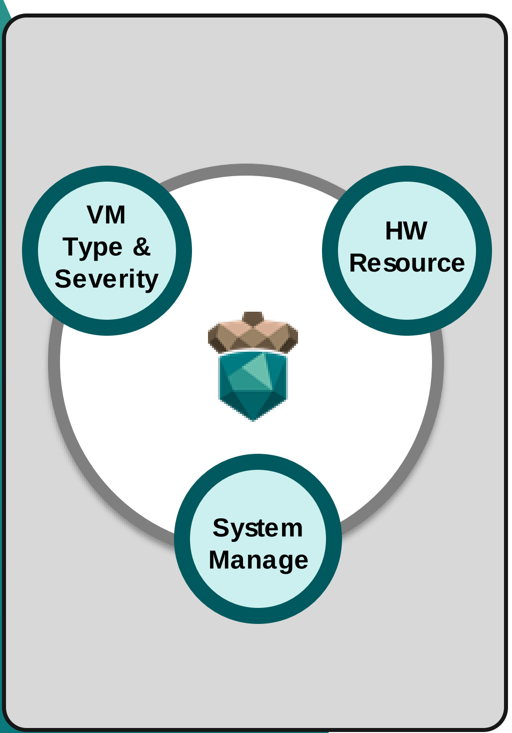
❑ No VMEXIT during runtime operations

❑ Highly optimized for RT:

- LAPIC passthrough
- RDT for resources isolation (cache and memory)
- PCI device passthrough
- Static CPU assignment

❑ Still use EPT and VT-d for VM isolation

System Manageability



VM Type & Severity:

- Load Order: Pre-launched, Service VM, Post-launched
- Category of VM: Service VM, User VM (can be pre-launched or post-launched)
- Severity: Safety VM > Hard RT VM > Soft RT VM > Service VM > Standard VM

HW Resource:

- CPU, memory & cache, devices, etc
- Partitioning or sharing based on VM type & severity

System Management:

- HW resources statically assigned at build time or dynamically assigned during runtime
- ACRN configuration tool: offline tool
- General reference design for VM & system lifecycle management
- Virtualization API: libvirt

System Security

Secure Boot

- Measured Boot
- Verified Boot

Isolation

- Isolation for mixed criticality workloads
- Intel Virtualization technologies: VT-x, VT-d
- Kata Containers
- EPT memory Isolation
- Interrupt isolation
- Cache Allocation Technology (CAT)

Runtime Security

- Virtual TPM
- Trusty
- Supervisor-Mode Access Prevention (SMAP)
- Supervisor-Mode Execution Prevention (SMEP)
- Software Guard Extension (SGX)
- Dynamic Application Loader (DAL)
- Total Memory Encryption (TME)

Rich I/O Mediation

- I/O device mediators

GPU	Ethernet	Block	Audio	IPU	I2C	GPIO	Touch	USB
SRIOV (*)	Virtio	Virtio	Virtio	Virtio	Virtio	Virtio	Virtio	Emu.

- Various security virtualization features

RPMB	CSE	TPM	Android Trusty	Verify Boot	Seed	SGX
Virtio	Virtio	Emu.	Emu.	Emu.	Emu.	Emu.

- PCI devices pass-through (VT-d) capability too

[*] Mediated Passthrough was supported on Intel Gen9 graphics

Diverse Guest OSes Supported

ubuntu
ANDROID
Real-Time LINUX
VxWorks
Windows 10
Zephyr Project
XENOMAI
yocto PROJECT

Ease of Use

Fast Development

- Short Learning Curve
- Straight-forward coding styles
- Multiple-channel Community (Mailing list, WeChat, TCM, etc)

Rich Documentation

- Getting Started Guide
- Architecture & Design
- Contributing Guides
- Tutorial
- Release notes

Easy Deployment

- Out-of-Box Experience
- VM Configuration Tool
- CPU assignment
- I/O sharing or pass-through
- Pre-defined Configuration
- Rich supported OS types
- Orchestration
- OTA

Flexible License

- BSD license for Hypervisor & Device models
- Dual Licenses for the ACRN Linux kernel drivers

Fin