



ResearchInChina
www.researchinchina.com

Global and China Automotive Operating System (OS) Industry Report, 2023-2024

Jan. 2024

Chinese operating systems start to work hard

In 2023, Chinese providers such as Huawei, Banma Zhixing, Xiaomi, and NIO made efforts in operating system market, launched different versions with competitive advantages, adapting to Chinese chip solutions and obtaining designated projects from OEMs.

Automotive Operating System Market Dynamics in China, 2023 (Partial)

Chinese Providers	Operating System	Type	Adapted Chinese Chip	Dynamics in 2023
Huawei	iDVP platform	SOA-based vehicle OS platform	-	Equipped with Luxeed S7, AITO M9
	Harmony	Cockpit OS	Kirin 990A/710A	Launched Harmony IVI 4.0, access to AI and Ark compiler on AITO M9 & AVATR 12
Banma Zhixing	AliOS	Cockpit-driving integrated OS	Horizon X3/J3/J5, SemiDrive X9H, etc.	Accessed to Tongyi Qianwen Large Model
CETC iSOFT Infrastructure Software	EASYADA	-	SemiDrive G9X	Joined China Automotive OS Open Source Plan and open-sourced microkernel EASYADA
NIO	SkyOS	Vehicle OS	Self-developed Shenji NX9031	Launched self-developed vehicle OS, equipped with ET9
ZTE	Automotive Microkernel OS	Intelligent driving OS	Black Sesame Technologies A1000	Launched fusion perception and reasoning solution
Geely	Flyme	Cockpit OS	SiEngine Long Ying No. 1	Equipped with Lynk & Co 08
Xiaomi	HyperOS	Vehicle OS	-	Equipped with Xiaomi SU7
ThunderSoft	DISHUI OS	Vehicle OS	-	Accessed to Large Model

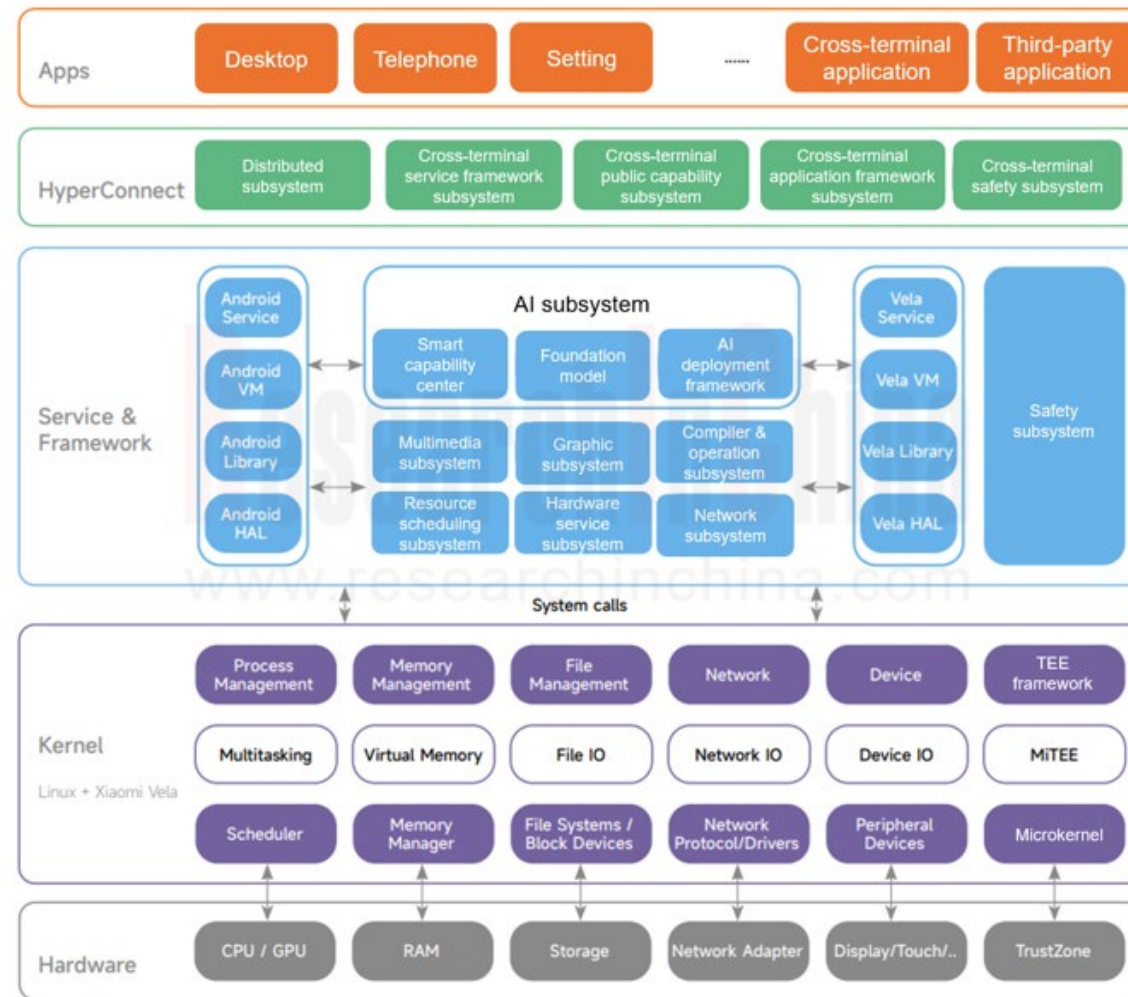
Source: ResearchInChina

For example, Xiaomi, NIO, etc. have launched vehicle operating systems, and providers such as iSOFT and ZTE have strengthened real-time, security and other functions of operating system microkernel.

In October 2023, Xiaomi launched its self-developed HyperOS, with the underlying layer uses fusion of Linux and Vela kernels, middleware access to AI subsystem, setting priorities in multitasking process, and adopts level scheduling to improve operating system processing efficiency.

In terms of ecosystem, Xiaomi has created the CarIoT ecosystem, which connects Internet IoT and automobile field, realizing concept of "full ecosystem of people, vehicle and home".

Xiaomi HyperOS Architecture

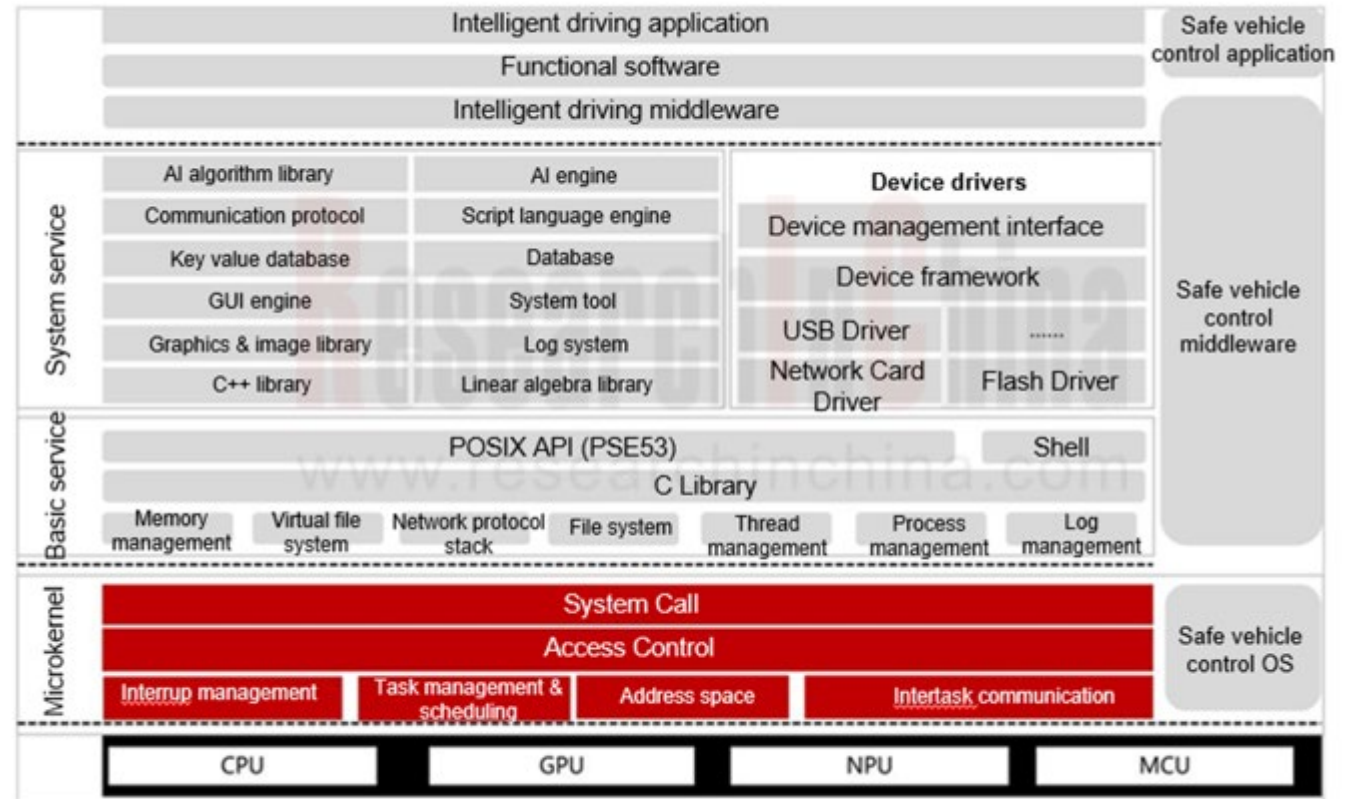


Source: Xiaomi

In middleware, AUTOSAR is still one of choices of most OEMs for vehicle control and autonomous driving. But it is not fully adapted to domestic chips, and feedback time of on-site communication processing is longer, which can't fully meet requirements of auto companies, resulting in partial providers and OEMs researching their own microkernel and middleware. In May 2023, China Association of Automobile Manufacturers (CAAM) formally released the first microkernel open-source project of China Automotive Operating System Open Source Plan, which plans to realize independent automotive OS based on open source microkernel and gradually replace QNX in 2025. Among them, iSOFT provided open source microkernel using the Mulan Public License (version 2); SemiDrive Technology provided G9X chip.

EasyAda microkernel provided by iSOFT can provide secure kernels for various chip platforms and application scenarios. For automotive field, iSOFT has implemented corresponding real-time improvement mechanisms for microkernel. For example, priority-based preemption mechanism for large-scale calculations of autonomous driving, preemption scheduling strategy of microkernel, as well as integrated algorithm, interrupt, delay mechanism and other technical means can improve microkernel to break through real-time and performance requirements.

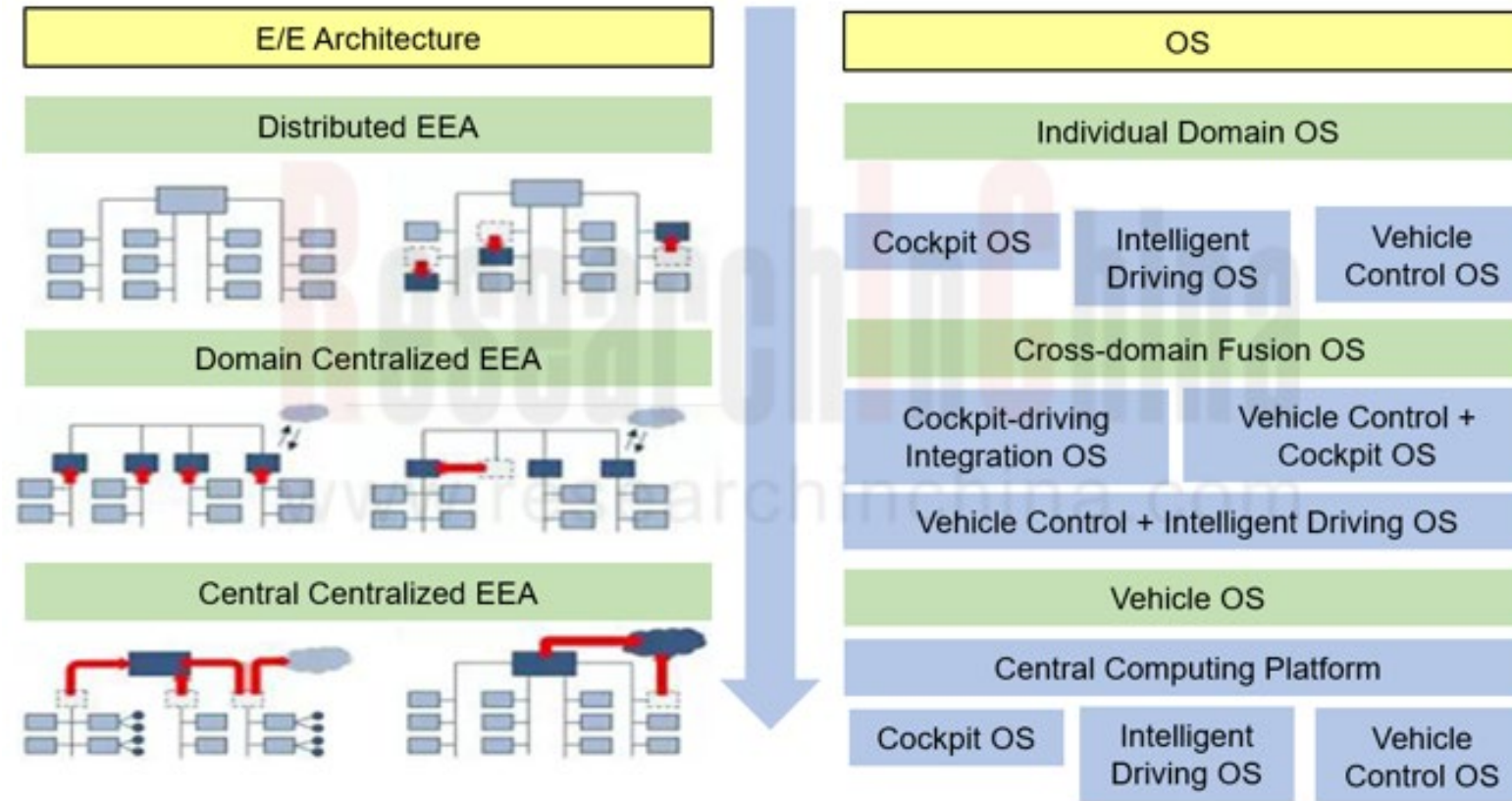
iSOFT's EasyAda Microkernel Architecture



Source: iSOFT

OEMs: three methods to realize vehicle operating systems

Automotive OS Evolution Route



Source: ResearchInChina

Three methods to implement vehicle operating system

As EE architecture evolves towards a centralized computing architecture, software systems begin to move towards an SOA architecture, where operating systems begin to progress from domain-type to vehicle-level.

SOA-based vehicle operating system integrates functions of cockpit OS, intelligent driving OS, and safety vehicle control OS via central computing platform to provide vehicle-level platform with a set of programming interfaces. Characterized by layered decoupling and unified architecture, it improves development efficiency of auto companies via providing unified interfaces. Vehicle operating system realizes functions scheduling and integration in various domains of the vehicle by centrally scheduling hardware resources and computing power.

According to ResearchInChina, there are three methods to implement vehicle operating system:

Three Methods of Vehicle Operating System Realization

Method	Vehicle OS Type	OEM Representative
1	Vehicle OS and chips are self-developed	AITO (Huawei), Tesla
2	<ul style="list-style-type: none">• Vehicle OS is self-developed;• Chips provided by vendors.	NIO, Xiaomi, Li Auto, Xpeng Motor, Leap Motor, etc.
3	<ul style="list-style-type: none">• Chip, underlying OS, middleware (partial) provided by vendors;• Middleware (partial), upper layer application self-developed.	Other OEMs

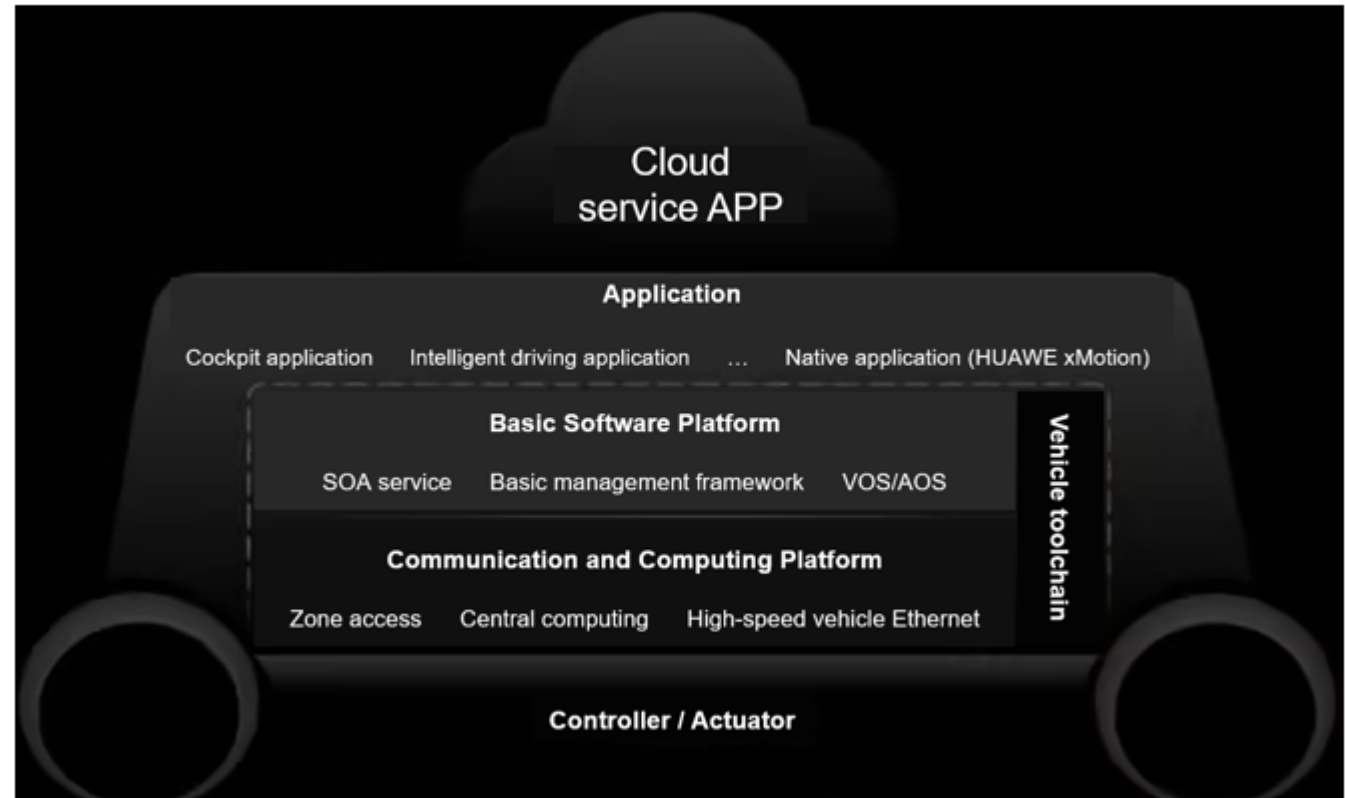
Source: ResearchInChina

Method 1, Huawei:

In April 2023, Huawei launched iDVP Intelligent Digital Vehicle Platform (i.e., vehicle operating system), which is a digital base designed based on SOA architecture, integrating functions of Huawei's various domain operating systems (AOS, HOS, and VOS), and realizing decoupling of software and hardware through atomic service layer, thus realizing rapid adaptation for cross-model development.

In 2023, representative model were AITO M9 and Luxeed S7, whose Toulin chassis is developed based on iDVP and realizes centralized and collaborative control of vehicle driving, braking, steering, and suspension through native applications such as HUAWEI xMotion configured with iDVP. In 2024, iDVP platform is scheduled to launch seven vehicles.

Huawei iDVP Solution



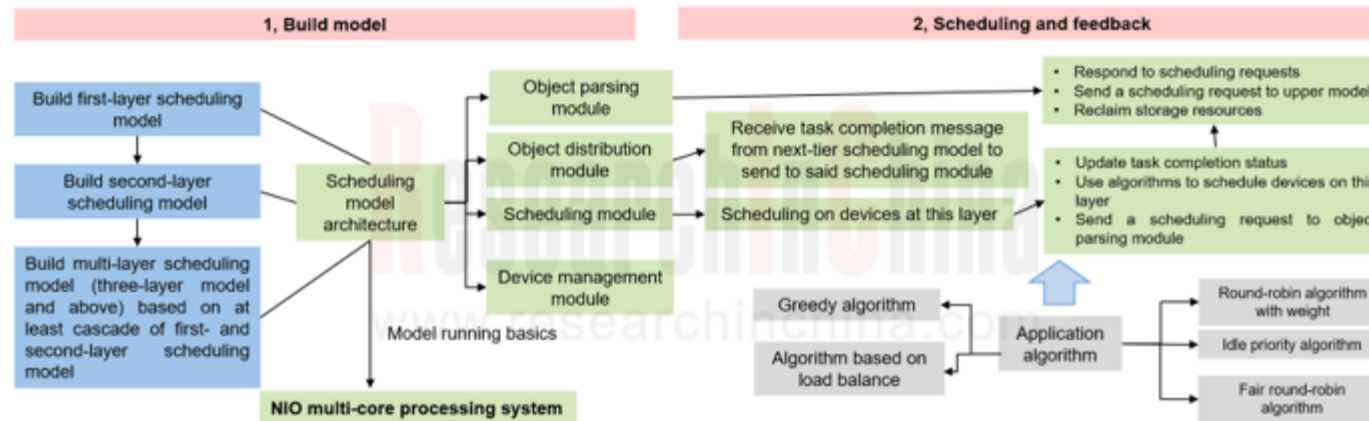
Source: Huawei

Method 2, NIO:

In September 2023, NIO released vehicle operating system "SkyOS", using self-developed microkernel and Hypervisor to replace QNX kernel service. SkyOS is divided into four modules, of which SkyOS-M module is based on a self-developed microkernel and has strong real-time performance and security. Its microkernel architecture is equipped with Hypervisor system for cockpit-driving integration, which is installed on NIO NT 3.0 platform; SkyOS-L module uses a self-developed middleware platform to replace AUTOSAR solution.

During OS development, NIO has released a number of technologies, including those for realizing task scheduling on multi-core processing system and improving task scheduling efficiency. Among them, multi-layer scheduling model is adopted for scheduling target tasks on multi-core processing system. equipped with fair round robin algorithm/most idle priority algorithm, etc., the vehicle operating system is able to coordinate hardware resources (computing power, sensors) under different working conditions by means of perception function groups.

Processes for Scheduling Target Tasks on Multi-core Processing System






Source: NIO

Comparison of Multitask Scheduling Methods in Linux, Harmony, and SkyOS

Linux, Harmony, SkyOS and other operating systems use different scheduling methods for multitasking, and evolved from initial unified scheduling to multi-layer scheduling, which improves processing efficiency, as well as security performance.

Comparison of Multitask Scheduling Methods in Linux, Harmony, and SkyOS

OS	Scheduling method	Scheduling feature
	Unified scheduling	No priority
	Level scheduling	Adopt dynamic monitoring + intelligent perception
	Multi-level scheduling	Scheduling model building

Source: ResearchInChina

Method 3, Volkswagen:

Volkswagen VW.OS consists of SDK (Software Development Kit), reference applications, software components and configuration tools for embedded software and cloud connectivity. By working in conjunction with VW.AC and BigLoop, it forms a vehicle software development platform that realizes conversion of distributed to centralized processing methods and achieves a core architecture reduction to three in-vehicle central processors.

As of February 2023, some Porsche and Audi models already carry partial components of VW.OS 1.2 (including software updates, cloud-based data transfer, diagnosis, and data accumulation), and Volkswagen plans to roll out the full software platform as version 2.0 in 2025, with partners including BlackBerry and Microsoft.

VW.OS Architecture



Source: Volkswagen

Providers: building an OS ecosystem

OS large-scale application requires the support of a strong ecosystem. In 2023, while actively implementing vehicle operating systems, automotive OS providers will also increase expansion of OS ecosystem, including adapting to more domestic chips and establishing more upstream and downstream partners.

Upstream/downstream cooperation:

ThunderSoft: worked closely with its subsidiary DISHUI Zhixing and Lingang section of Shanghai Pilot Free Trade Zone to set up a vehicle R&D base; in addition, ThunderSoft and Cariad, a subsidiary of Volkswagen, established Carthunder as a joint venture to cooperate in the fields of intelligent connectivity, intelligent cockpit, and operating system.

ArcherMind Technology: established a strategic partnership with EB in AUTOSAR.

Chip:

ThunderSoft: ThunderSoft is deeply bound to Qualcomm chips. For example, it has achieved stable operation of the LLaMA-2 13 billion parameter model on edge devices equipped with Qualcomm 8 series chip platforms, and improved competitiveness of vehicle platform products through AI large models.

ArcherMind Technology: In 2023, ArcherMind Technology built a Hesper OS software platform solution for J5 and J6 based on Horizon TogetherROS.Auto platform. This solution adds SOA functions based on FusionDrive functions. In addition, ArcherMind Technology signed a cooperation agreement with NVIDIA to become its ecosystem software partner, providing intelligent driving vision solutions based on Orin and Xavier chips to intelligent driving-related companies.

Build a developer ecosystem: Huawei, for example, launched HarmonyOS NEXT and provides middleware and tool chains for developers.

Development History of ThunderSoft Vehicle OS



Source: ThunderSoft

Table of Content (1)

1 Trends of Automotive OS

1.1 Trend 1: Changes in Underlying OS Competitive Landscape

1.1.1 Market Status

1.1.2 Underlying OS Market News

1.1.3 WinCE Officially Exits the Market

1.1.4 Linux Potential Competitiveness Strengthening

1.1.5 Domestication Status

1.2 Trend 2: Operating System Toward Open Source Integration Stage

1.2.1 China Automotive OS Open Source Plan

1.2.2 Key to Open Source Integration

1.3 Trend 3: Vehicle OS under SOA Architecture

1.3.1 Definition of Vehicle OS

1.3.2 Software Layer Architecture

1.3.3 Characteristics of Vehicle OS

1.3.4 Evolution of Vehicle OS Development Modes

1.3.5 Evolution of Vehicle OS Business Models

1.3.6 Vehicle OS Layout of OEMs

1.3.7 Vehicle OS Layout of Providers

1.3.8 Vehicle OS Cross-domain Scheduling: Algorithm Call/Chip Adaptation

1.3.9 Integration of Vehicle OS with AI Large Model

1.4 Trend 4: OEMs Self-developed OS

1.4.1 Advantages and Risks

1.4.2 Current Status

1.4.3 Solution Comparison

1.5 Trend 5: IVI System

1.5.1 Weakening of Mapping Software Function

1.5.2 Changes in Market Share of Leading Providers

2 Overview of Automotive OS

2.1 Introduction of Automotive OS

2.1.1 Definition of Automotive OS

2.1.2 History of OS Installation

2.2 Automotive OS Classification

2.2.1 Automotive OS Classification: Narrow-sense OS and Broad-sense OS

2.2.2 Automotive OS Classification: Real-time and Non-real-time

2.2.3 RTOS Suppliers and Product List

2.2.4 Non-RTOS Suppliers and Product List

2.2.5 Automotive OS Classification: Micro Kernel, Macro Kernel, and Hybrid Kernel

2.2.6 Automotive OS Classification: Vehicle Control Operating System and In-vehicle Operating System

2.2.7 In-vehicle OS Industry Map

2.2.8 Automotive OS Market Size

2.3 Software Architecture

2.3.1 Intelligent Vehicle Software Architecture

2.3.2 Intelligent Vehicle Software Ecological Framework

2.3.3 Kernel is the Core of Automotive Software Architecture

2.4 Business Model

2.4.1 Business Model Types

2.4.2 Business Models of Major Automotive Operating System Enterprises

Table of Content (2)

2.4.3 Automotive Operating System Development Trend and Business Model
Exploration

2.4.4 Basic Automotive Operating System and Business Model

2.4.5 Automotive RTOS and Business Model

2.4.6 Suppliers OS and Business Model

2.5 Basic Automotive OS

2.5.1 Introduction

2.5.2 Automotive Underlying OS Market Share

2.6 Customized Automotive OS

2.6.1 Introduction

2.6.2 Comparison

2.6.3 Chip Vendors and Their Partners

2.7 ROM Automotive OS

2.8 Hypervisor

2.8.1 Introduction

2.8.2 Hypervisor Becomes an Inevitable Choice

2.8.3 Comparison of Main Hypervisors

2.9 OSEK

2.9.1 Introduction

2.9.2 Architecture and Features

2.10 Open Organization: Covesa

2.10.1 Profile

2.10.2 Members

2.10.3 Achievements

2.10.4 Achievement Examples

2.10.5 Role of Covesa

2.10.6 Dynamics

2.11 Open Organization: AUTOSAR

2.11.1 Profile

2.11.2 Classification

2.11.3 Members

2.11.4 Classic AUTOSAR

2.11.5 Adaptive AUTOSAR

2.11.6 Comparison of Classic and Adaptive AUTOSAR

2.11.7 Integration of Adaptive AUTOSAR and ROS

2.11.8 Core of AUTOSAR

2.11.9 Organization of AUTOSAR China Working Group

2.11.10 Project Case of AUTOSAR China Working Group

2.11.11 AUTOSAR Related Software Tool Supplier Business Model

2.11.12 Vector AUTOSAR Solution Business Model

2.11.13 EB AUTOSAR Solution Business Model

2.11.14 Neusoft Reach AUTOSAR Solution Business Model

2.11.15 Business Model of iSOFT AUTOSAR Solution

2.11.16 Business Model of Jingwei HiRain AUTOSAR Solution

2.12 Open Organization: Autoware Foudation

2.12.1 Profile

2.12.2 Features of Autoware Core

3 Basic Automotive OS

Table of Content (3)

- 3.1 QNX
 - 3.1.1 Profile
 - 3.1.2 Business
 - 3.1.3 QNX Real-time Operating System (Neutrino RTOS)
 - 3.1.4 QNX Neutrino RTOS Support Platform
 - 3.1.5 QNX Vehicle Application
 - 3.1.6 QNX Cockpit Software Platform Solution
 - 3.1.7 Platform for ADAS
 - 3.1.8 Safety Level
 - 3.1.9 OS for Safety
 - 3.1.10 Cockpit-driving Integrated Domain Controller
 - 3.1.11 Partners
 - 3.1.12 Dynamics
- 3.2 Linux&AGL
 - 3.2.1 Profile
 - 3.2.2 AGL Architecture
 - 3.2.3 Linux Architecture
 - 3.2.4 Iteration History
 - 3.2.5 OEM Partners
 - 3.2.6 RT-Linux
 - 3.2.7 Members
 - 3.2.8 Dynamics
- 3.3 Android
 - 3.3.1 Profile of Android & Android Automotive OS
 - 3.3.2 Android Automotive OS Features
 - 3.3.3 Underlying Architecture
 - 3.3.4 User Development

- 3.3.5 Dynamic News
- 3.4 VxWorks
 - 3.4.1 Profile
 - 3.4.2 WindRiver Products: VxWorks
 - 3.4.2 WindRiver Products: WindRiver Linux and Wind River AUTOSAR Adaptive Software Platform
 - 3.4.2 WindRiver Products: Helix Virtualization Platform
 - 3.4.3 WindRiver VxWorks Microkernel Architecture
 - 3.4.4 Main Partners in Automotive Field
 - 3.4.5 Wind River's Dynamics in Automotive Field
 - 3.4.6 New Application: Aptiv Introduces Wind River Studio "Cloud Native" to Automotive Industry
- 3.5 AliOS
 - 3.5.1 Profile
 - 3.5.2 Architecture
 - 3.5.3 Application Layer
 - 3.5.4 Solutions
 - 3.5.5 AliOS Drive
 - 3.5.6 Cooperation between Banma Zhixing and AliOS Drive
 - 3.5.7 AliOS Accessed to Large Model
 - 3.5.8 Models Equipped
 - 3.5.9 Dynamic News
- 3.6 Harmony OS
 - 3.6.1 Profile
 - 3.6.2 Development History
 - 3.6.3 Technical Features
 - 3.6.4 Cooperation Model with Automakers

Table of Content (4)

- 3.6.5 Cockpit HOS
- 3.6.6 Intelligent Driving OS: AOS
- 3.6.7 Intelligent Vehicle Control OS: VOS
- 3.6.8 Huawei Cross-domain Integrated Software Framework
- 3.6.9 CCA Architecture: VCU + 3-5 VIUs
- 3.6.9 CCA Architecture: System Framework and Full-stack Solution
- 3.6.10 Harmony Access to Large Models
- 3.6.11 Models Equipped
- 3.6.12 Dynamic News

- 3.7 Ubuntu
- 3.7.1 Profile
- 3.7.2 Historical Versions
- 3.7.3 Application
- 3.7.4 Cooperation in Automotive Field

- 3.8 ROS
- 3.8.1 Profile
- 3.8.2 Introduction to ROS 2.0
- 3.8.3 ROS 2.0 Iteration
- 3.8.4 ROS 2.0 Architecture
- 3.8.5 Application Cases

- 3.9 webOS
- 3.9.1 Development History
- 3.9.2 OSE Components and Development Roadmap
- 3.9.3 Integration with AGL
- 3.9.4 Dynamics in Automotive Field

4 Hypervisor

- 4.1 Profile
- 4.1.1 Definition
- 4.1.2 Comparison of Main Hypervisors
- 4.1.3 Status Quo of Hypervisor Industry
- 4.1.4 Application of Intelligent Cockpit Hypervisors in China
- 4.1.5 Prospects of Global Automotive Hypervisor Market
- 4.1.6 Global Hypervisor Suppliers and Their Product Lists
- 4.1.7 Chinese Hypervisor Suppliers and Their Product Lists
- 4.1.8 Automotive Virtual Machine Management System Business Model

- 4.2 QNX Hypervisor
- 4.2.1 Profile
- 4.2.2 Architecture
- 4.2.3 Solutions

- 4.3 ACRN
- 4.3.1 Profile
- 4.3.2 Composition

- 4.4 COQOS Hypervisor
- 4.4.1 COQOS Hypervisor
- 4.4.2 COQOS Hypervisor SDK
- 4.4.3 Mixed VIRTIO / Non-VIRTIO Architectures
- 4.4.4 “Next Gen COQOS” Heterogeneous Cores

- 4.5 PikeOS
- 4.6 EB Corbos Hypervisor
- 4.7 Harman Device Virtualization

Table of Content (5)

- 4.8 VOSYSmonitor
- 4.9 ZlingSmart Hypervisor

5 Broad-sense Automotive OS and Enterprises

- 5.1 Neusoft Reach NeuSAR
 - 5.1.1 Intelligent Connected Vehicle Business Layout
 - 5.1.2 Introduction to NeuSAR
 - 5.1.3 Dynamics of NeuSAR
 - 5.1.4 Main Products
 - 5.1.5 Vehicle Operating System
 - 5.1.6 Neusoft Reach Domain Controller Software Development Platform: NeuSAR DS
 - 5.1.7 NeuSAR Security Levels
- 5.2 ThunderSoft
 - 5.2.1 Profile
 - 5.2.2 Business
 - 5.2.3 DISHUI OS
 - 5.2.4 Combination between Rubik Large Model and OS
 - 5.2.5 OS Projects
 - 5.2.6 SOA Middleware
- 5.3 ETAS
 - 5.3.1 Vehicle Solutions
 - 5.3.2 Cloud Native Software Platform
- 5.4 ArcherMind Technology
 - 5.4.1 Profile

- 5.4.2 Main Products
- 5.4.3 Subsidiary: Arraymo
- 5.4.4 Operating System: FusionOS
- 5.4.5 Cooperate with Horizon on Intelligent Driving Domain OS Platform
- 5.4.6 Fusion Matrix
- 5.4.7 FusionDrive: Some IP Products and Smart Gateways
- 5.4.8 Architecture for Some IP Integration with TogetheROS.Auto
- 5.4.9 J5-based SomeIP Fusion Architecture
- 5.4.10 J5-based FusionDrive Diagnosis
- 5.4.11 Partners
- 5.5 EB
 - 5.5.1 Profile and Products Introduction
 - 5.5.2 J5-based Intelligent Driving Domain OS
- 5.6 AICC
 - 5.6.1 Profile
 - 5.6.2 AICC Promotes the Automotive OS Standard
 - 5.6.3 ICVOS: Intelligent Connected Vehicle OS
 - 5.6.4 ICVOS: Software Architecture /Development Architecture /SDK Architecture
 - 5.6.5 ICVOS: Vehicle-cloud Collaboration
 - 5.6.6 ICVOS: Basic Information Security Platform
 - 5.6.7 ICVOS: Software Architecture Development Case with OEM
- 5.7 NVIDIA DRIVE OS
 - 5.7.1 Profile
 - 5.7.2 Drive OS SDK Architecture
- 5.8 Baidu DuerOS

Table of Content (6)

- 5.8.1 Profile
- 5.8.2 Features
- 5.8.3 Application Scenarios and Customers
- 5.8.4 Software Architecture
- 5.8.5 Baidu DuerOS Accesses to Large Model
- 5.8.6 Baidu Vehicle OS Plan

- 5.9 Apex.AI
 - 5.9.1 Profile
 - 5.9.2 Dynamics
 - 5.9.3 Apex.OS Features
 - 5.9.4 Apex.OS Extension Kit
 - 5.9.5 Apex.OS Tool Architecture

- 5.10 iSOFT Infrastructure Software
 - 5.10.1 Profile
 - 5.10.2 Development Course
 - 5.10.3 Product and Service
 - 5.10.4 AUTOSAR CP+AP Integrated Solution
 - 5.10.5 OS Architecture
 - 5.10.6 Vehicle OS Architecture
 - 5.10.7 Orientais
 - 5.10.8 Open-source Microkernel

- 5.11 ZlingSmart
 - 5.11.1 Profile
 - 5.11.2 RAITE OS

- 5.12 ZTE GoldenOS

- 5.12.1 Micro kernel and Macro kernel Technology Architecture
- 5.12.2 Vehicle Control OS Solution
- 5.12.3 Intelligent Cockpit OS Solution
- 5.12.4 Intelligent Driving OS Solution
- 5.12.5 Cooperation of Neusoft Reach + ZTE + SemiDrive Technology
- 5.12.6 Introduction of NewStart OS
- 5.12.7 Development History of NewStart
- 5.12.8 NewStart's Intelligent Cockpit OS Solution

- 5.13 ECARX
 - 5.13.1 Product Introduction
 - 5.13.2 Ecological Cooperation

- 5.14 Others
 - 5.14.1 RT-Thread
 - 5.14.2 Enjoy Move Technology: EMOS
 - 5.14.3 KOTEI Information: Kcar-OS
 - 5.14.4 PATEO Qing OS
 - 5.14.5 STEP

6 Vehicle OS of OEMs

- 6.1 Tesla OS
 - 6.1.1 Introduction to Tesla OS
 - 6.1.2 Tesla: Cloud OS

- 6.2 NIO SkyOS
 - 6.2.1 Introduction
 - 6.2.2 SkyOS Architecture

Table of Content (7)

6.2.3 NIO Vehicle OS Scheduling Algorithm

6.2.4 Adapted Chips

6.3 Xiaomi Hyper OS

6.3.1 Introduction

6.3.2 OS Development History

6.3.3 Architecture Design

6.3.5 Access to Large Model

6.3.6 Adopting Nuttx Kernel

6.4 Li Auto

6.4.1 Li OS Architecture

6.4.2 Self-developed Chips Status

6.5 Xpeng Motor XsmartOS

6.5.1 X-EEA2.0 Architecture

6.5.2 X-EEA3.0 Key Technologies

6.5.3 Next Steps

6.6 Leap Motor

6.6.1 Four-domain Fusion Architecture

6.6.2 Intelligent Driving OS

7 OS of Other OEMs

7.1 Volkswagen VW.OS

7.1.1 Introduction

7.1.2 Development History

7.1.3 Features

7.1.4 Overall Layout of Software

7.1.5 Roadmap - Hardware, Software to be Unified

7.1.6 E3 2.0 EEA: CARIAD Software Platform

7.2 Toyota Arene OS

7.2.1 Introduction

7.2.2 Ecological Resources

7.2.3 Functions

7.3 BMW

7.3.1 Mass-produced EEA: Software System Development Evolution

7.3.2 BMW iDrive

7.4 Mercedes-Benz MB OS

7.4.1 Function Introduction

7.4.2 Architecture

7.4.3 Recent Developments

7.5 FAW.OS

7.5.1 Architecture

7.5.2 Features

7.6 SAIC Z-ONE Intelligent Vehicle Cloud-Management-End Full Stack Solution-SOA Software Platform

7.7 GAC Psi OS

7.7.1 Architecture

7.7.2 Application

Table of Content (8)

7.8 Geely

7.8.1 VolvoCars.OS

7.8.2 Flyme OS

7.8.3 SOA-based OS - GeelyOS

7.8.4 ZEEKR OS

7.9 Great Wall Motor

7.9.1 GEEP 4.0 Quasi-Central EEA: SOA Software Framework

7.9.2 Coffee OS

7.9.3 GEEP 5.0 Central Computing EEA

7.10 Changan RTDriveOS

7.10.1 Architecture

7.10.2 Next Steps

7.11 CHERY-OS

7.11.1 Architecture

7.11.2 Applications

7.12 Dongfeng

7.12.1 Vehicle OS Architecture

7.12.2 Vehicle OS Development Route

7.13 BYD OS



Beijing Headquarters

TEL: 13718845418

Email: report@researchinchina.com

Website: [ResearchInChina](http://ResearchInChina.com)

WeChat: Zuosiqiche



Chengdu Branch

TEL: 028-68738514

FAX: 028-86930659

