



# 万物互联时代的操作系统

## 研究报告

亿欧智库 <https://www.iyiou.com/research>

Copyright reserved to EO Intelligence, October 2020

## 研究背景与目的

物联网因其宏大主题，以及巨大的应用想象空间而备受政府和企业的关注。**国际电信联盟 (ITU) 对物联网的定义**：“物联网主要解决物品到物品 (Thing to Thing, T2T)，人到物品(Human to Thing, H2T)，人到人 (Human to Human, H2H) 之间的互联。”因此物联网也被视为是继计算机、互联网之后的第三次技术革命。但由于物联网包含的软硬件技术太多，既牵涉到硬件端如传感器、通讯模组、异构芯片，也涉及到软件层如操作系统和应用软件，行业发展遇到不少困境，行业整体呈波段式前进。

AIoT+云计算+5G的发展，使得万物互联正在逐渐成为现实，整个ICT产业可能有一次重塑的机会。数据的连接节点、传输速度、时延、处理的规模和速度得到不断的提升，“万物互联”的AIoT将会在智能家居、智慧工厂、智慧城市等场景下均有较大表现。

**亿欧智库**一直关注于物联网产业的发展，自2018年发布《2018中国物联网应用研究报告》以来，后续又推出了深入探讨物联网技术发展及应用的报告。《2019年中国智慧城市发展研究报告》、《2020中国车联网商业模式分析报告》、《2020智能物流产业研究报告》、《2020巨头企业物联网业务布局研究》等已发布报告从不同角度探讨物联网技术的发展。

我们也越来越关注底层技术在物联网产业发展中起到的推动力。操作系统作为重要底层软件技术之一，上承应用开发，下接海量的硬件终端，正在吸引巨头提前布局。我们观察到国内外科技巨头都在操作系统层面有所布局，比如谷歌的Android Things和Fuchsia、阿里的AliOS和AliOS Things、华为的LiteOS和鸿蒙OS、ARM的MbedOS等。

但市场中对于面向万物互联时代的操作系统还未有充分的讨论，我们希望借助这份报告来实现对物联网操作系统 (IoT OS) 的定义与梳理，提供给物联网从业人士和其他对物联网产业感兴趣的人士一些有用见解。

## 主要研究发现

- 物联网终端海量碎片化的特征，很可能使得物联网的操作系统并不像PC或手机系统具有个别企业垄断市场的局面；

桌面端：Windows系列 88.23% + Mac系列 8.9% = 97.13%

移动端：Android 74% + iOS 25%=99%



海量碎片的物联网端：**分业务、分场景、分生态**

- 云计算厂商和终端硬件设备厂商有充分的动力去推动自研物联网操作系统，“云+端”是物联网操作系统的主要竞争场；云计算厂商做IoT OS的代表有阿里云、谷歌云、AWS、Azure等，硬件相关厂商做IoT OS的代表有华为、ARM、苹果等。技术积累和资源禀赋决定了不同厂商的入局点不同。
- 当前阶段应用的物联网操作系统往往呈现出轻量级系统和通用级系统并存的情况，轻量级满足低计算资源（RAM及ROM资源最小以KB计）场景，通用级适用于计算资源较好（RAM及ROM资源在MB以上，甚至达到GB）的物联网场景，是否会出现一种统一的IoT OS，还是多套系统的配合实现功能，目前而言下定论还为时尚早。巨头尝试去打破操作系统这种异构状态，但前景不明朗。新老操作系统的交替会持续相当一段时间。
- 嵌入式操作系统几乎每10年会发生一次重大模态进化，目前的时间节点下来比较，很类似于2005-2008年移动操作系统的爆发前夜，当时无论是微软、惠普&Palm、诺基亚、谷歌、三星、黑莓，还是苹果公司都在积极探索面向下一代的便携式个人计算机新形态，并最终出现智能手机的繁荣。

- **MCU**：微控制单元（Microcontroller Unit），又称单片微型计算机（Single Chip Microcomputer）或者单片机，是把中央处理器（CPU）的频率与规格做适当缩减，并将内存（memory）、计数器（Timer）、USB、A/D转换、UART、PLC、DMA等周边接口，甚至LCD驱动电路都整合在单一芯片上，形成芯片级的计算机，为不同的应用场合做不同组合控制。
- **SoC**：系统级芯片或片上系统（System on a Chip），国内外学术界一般倾向将SoC定义为将微处理器、模拟IP核、数字IP核和存储器（或片外存储控制接口）集成在单一芯片上，它通常是客户定制的，或是面向特定用途的标准产品。
- **嵌入式系统**（Embedded system）：以应用为中心、计算机系统为基础，软硬件可裁剪，适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。
- **操作系统**（Operating System，简称OS）：管理计算机硬件和软件资源的基础软件。操作系统需要处理内存管理、配置系统资源、控制输入输出设备、操作网络与管理文件系统等基本事务。
- **实时操作系统**（Real Time Operating System，简称RTOS）：操作系统的一种，系统保证在一定时间限制内完成。实时操作系统有硬实时和软实时之分，硬实时要求在规定的时间内必须完成操作；软实时则只要按照任务的优先级，尽可能快地完成操作即可。
- **物联网操作系统**（IoT OS）：目前对于IoT OS并无明确的统一定义，可以认为其是在嵌入式基础上，面向物联网开发和功能实现需求（低功耗、通讯协议、二次开发、云端对接、高可靠等），使嵌入式系统具备网络连接和通信能力的操作系统总称。
- **宏内核**（Monolithic kernel）：操作系统核心架构的一种，特性是整个核心程序都是以核心空间（Kernel Space）的身份及监管者模式（Supervisor Mode）来运行。用户服务和内核服务运行在同一地址空间中。

- **微内核** (Micro kernel) : 仅提供操作系统的最核心功能, 如进程调度、进程间通信、存储管理等, 而诸如文件管理、网络支持等服务以接口形式和微内核通信。因此与宏内核相比尺寸小、可扩展性好。
- **进程间通信** (Interprocess communication, IPC) : 程序运行中, 每个进程各自有不同的用户地址空间, 任何一个进程的全局变量在另一个进程中都看不到, 所以进程之间要交换数据必须通过内核, 在内核中开辟一块缓冲区, 进程1把数据从用户空间拷到内核缓冲区, 进程2再从内核缓冲区把数据读走, 这一过程称为进程间通信。
- **系统总线** (System Bus) : 系统总线是连接计算机系统的主要组件, 包括数据总线 (Data Bus)、地址总线 (Address Bus) 和控制总线 (Control Bus) 。
- **随机存取存储器** (Random Access Memory, RAM) : 是与CPU等计算单元交换数据的内部存储器。作为操作系统或其他正在运行中程序的临时数据存储介质。RAM具有数据易失性, 在计算机系统中暂时存储程序、数据和中间结果, 断电所存储的数据将随之丢失。
- **只读存储器** (Read-Only Memory, ROM) : ROM存储数据稳定, 断电后所存数据也不会改变, 并且结构较简单, 常用于存储各种固定程序和数据。
- **可移植操作系统接口** (Portable Operating System Interface, POSIX) : POSIX标准定义了操作系统应该为应用程序提供的接口标准, 是IEEE为要在各种UNIX操作系统上运行的软件而定义的一系列API标准的总称。
- **软件开发工具包** (Software Development Kit, SDK) : 为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件时的开发工具集合。

- **应用程序接口** (Application Programming Interface, API) : 一组定义、程序及协议的集合, 通过 API接口实现计算机软件之间的相互通信。开发者无需理解软件内部工作的机制细节, 也无需访问源码即可访问程序实现功能。API可降低系统各部分的相互依赖, 提高组成单元的内聚性, 降低耦合度。
- **集成开发环境** (Integrated Development Environment, IDE) : IDE为程序开发提供开发环境, 包括代码编辑器、编译器、调试器和用户图形界面等工具, 集成有代码编写功能、分析功能、编译功能、调试功能等一体化开发套件。
- **中间件** (Middleware) : 中间件是独立的系统级软件, 连接操作系统层和应用程序层, 将不同操作系统提供应用的接口标准化, 协议统一化。
- **硬件抽象层** (Hardware Abstraction Layer, HAL) : 介于操作系统内核与硬件电路之间的接口层, 目的在于将硬件抽象化。在隐藏了特定平台的硬件接口细节后, 为操作系统提供虚拟硬件平台, 使其与硬件无关, 实现多硬件平台移植的目的。
- **图形用户界面** (Graphical User Interface, GUI) : 以图形化形式提供用户与计算机之间的交互, 由相应硬件平台和操作系统来支持显示。
- **谷歌移动服务** (GoogleMobile Service, GMS) : GMS是Google开发并推动Android的动力, 为谷歌play所应用的底层框架, 闭源运行。GMS提供包括GooglePlay、Search、Gmail、Talk、Maps、Street View、YouTube、Android Market等服务。海外Android平台严重依赖GMS。
- **安卓开放源代码项目** (Android Open-Source Project, AOSP) : 安卓项目的开放源代码部分, 对应于GMS。

# 目录

## CONTENTS

### 1. 万物互联时代的战略制高点

- 1.1 物联网操作系统的时代背景与使命挑战
- 1.2 物联网操作系统的市场空间

### 2. 操作系统历史经验回顾

### 3. 物联网操作系统代表类型及案例

- 3.1 嵌入式操作系统
- 3.2 通用OS裁剪后的操作系统
- 3.3 轻量级物联网操作系统
- 3.4 下一代操作系统

### 4. 物联网操作系统发展展望

# 一、万物互联时代的战略制高点

The commanding heights of IoT

# 信息化发展已从计算机时代、互联网时代发展到物联网时代， 联网要求由“人人相连” 变为 “人物交互”， 物联网产业处于快速成长期

20世纪60年代，美国四所大学的超级计算机开始采用TCP/IP协议联机，正式拉开了互联网时代的序幕。经过60余年的发展，设备网络互连已从大型计算机时代发展到个人计算机时代，**我们现在正处于移动计算时代和万物互联时代的过渡阶段**，软件及硬件设备发生了翻天覆地的变化，联网的要求由之前的“人与人”相连转变为“人与人”及“人与物”相连。近十年来，全球PC和智能手机的出货量为百万级别，而物联网的设备连接量达到了百亿级别，**物联网因其应用广泛性、用户友好性、生态完整性而具有巨大的发展潜力。**



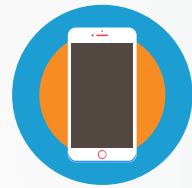
## 大型计算机时代

**时代特点**  
有线互联，算力强大  
用于科研、企事业单位



## 个人计算机时代

有线/无线互联  
互联网开始普及



## 移动计算机时代

无线互联，实时连接  
终端设备平民化



## 边缘计算机时代

万物互联，AI赋能  
全场景式智能应用

通讯技术实现了由1G向5G的跨越

### 代表公司

IBM  
雷明顿-兰德

IBM Apple 联想  
惠普 戴尔 宏碁

华为 Apple  
Samsung 小米

华为 阿里巴巴 小米  
Microsoft Apple Amazon

### 设备连接量

数量级别为  
**千万台**

2010年全球出货量峰值  
**3.5亿台**

2016年全球出货量峰值  
**14.7亿台**

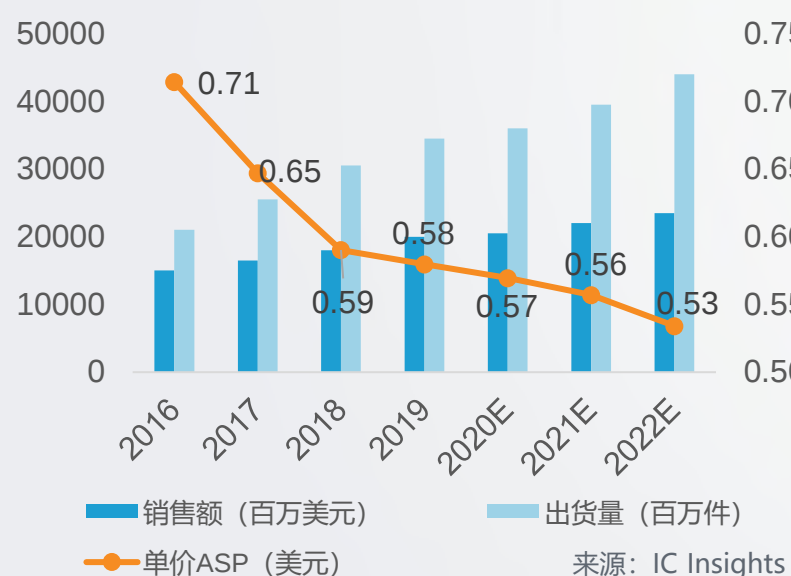
2019年全球设备连接量  
**120亿台**

# 硬件更迭和价格下降推动终端生态的普及，进一步推动物联网产业的发展

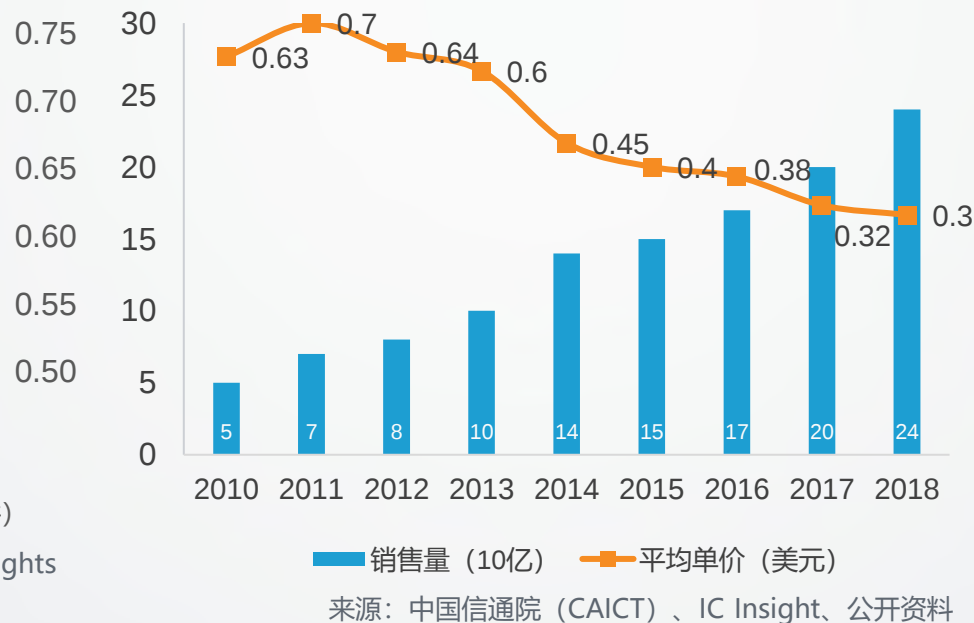
物联网的海量终端上，硬件成本问题永远是重大考验之一。微控制器（MCU）、MEMS传感器和无线通讯模组成本的逐年下降从成本端推动行业发展。MCU在性能提升和价格下降的道路上不断前进，寄存器位数从4位、8位、16位到今天主流的32位MCU，主频则从8MHz、16MHz发展到现在的几百MHz。据IC Insights预计，MCU的平均单价在未来3年内会逐渐下降。**价格不断下降的高性能MCU将为物联网应用提供更多可能。**

传感器和无线通讯模组行业的发展呈现出相似的趋势。NB-IoT模组及Wi-Fi模组的价格在近年发生剧烈下降，分别自5年前的几十元下降到近期的10-20元附近，成为中远距离和近距离场景的最经济选择。**全球传感器平均单价持续走低，自2010年以来下降幅度超过100%。**

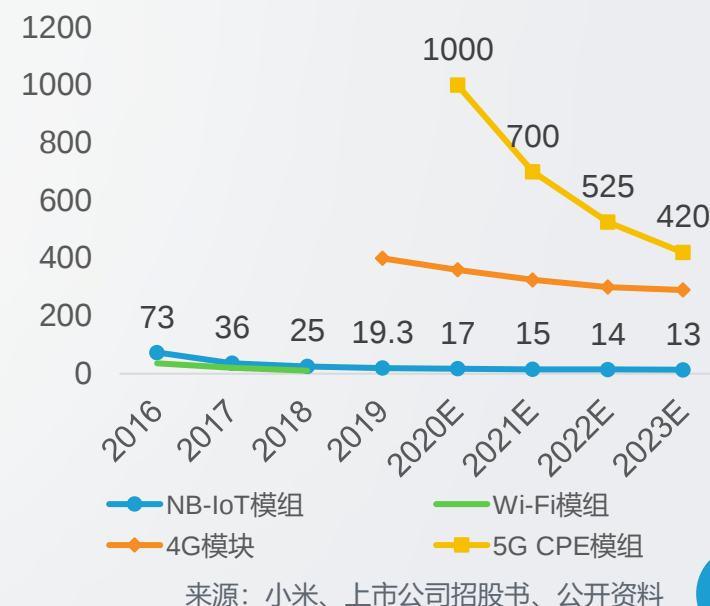
## 亿欧智库：MCU的平均单价逐年递减



## 亿欧智库：2010-2018年传感器价格逐年下降



## 亿欧智库：物联网通讯模组价格预期变化

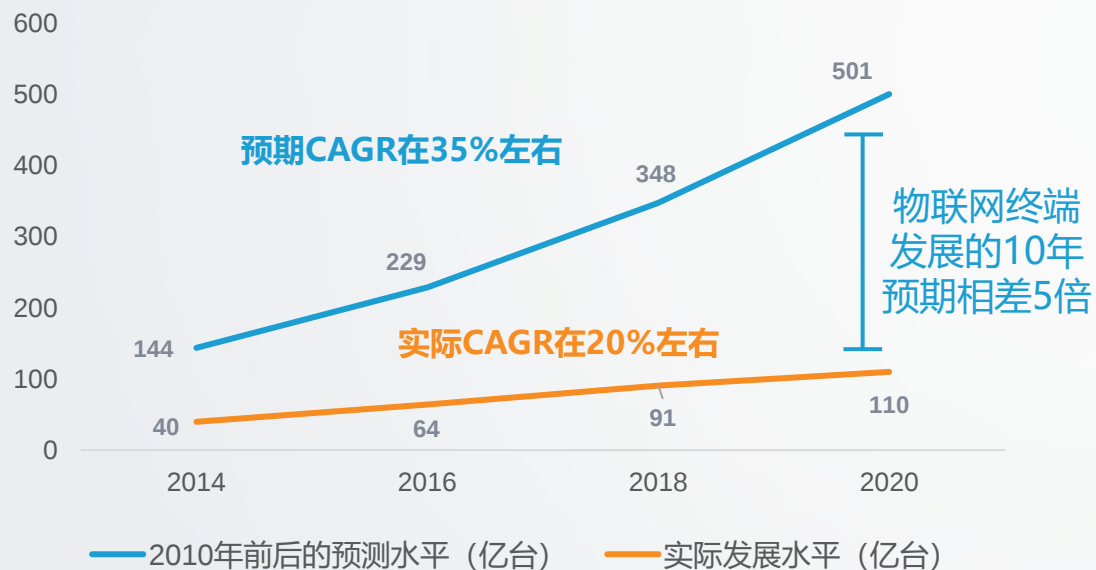


# 物联网十年发展不及预期，软硬件紧密耦合带来生态壁垒，物联网操作系统是破局关键

从1982年的第一台联网设备开始，物联网已经走过了近40年。在2010年左右，爱立信、IBM、思科、Gartner对物联网产业的发展相当乐观，普遍预测至2020年全球将有500亿台联网设备。但时间真的来到2020年，我们发现联网设备数量这一最关键指标的达成率低于预期，Gartner给出的2020年预计120亿连接数量，远低于10年前的预期水平。

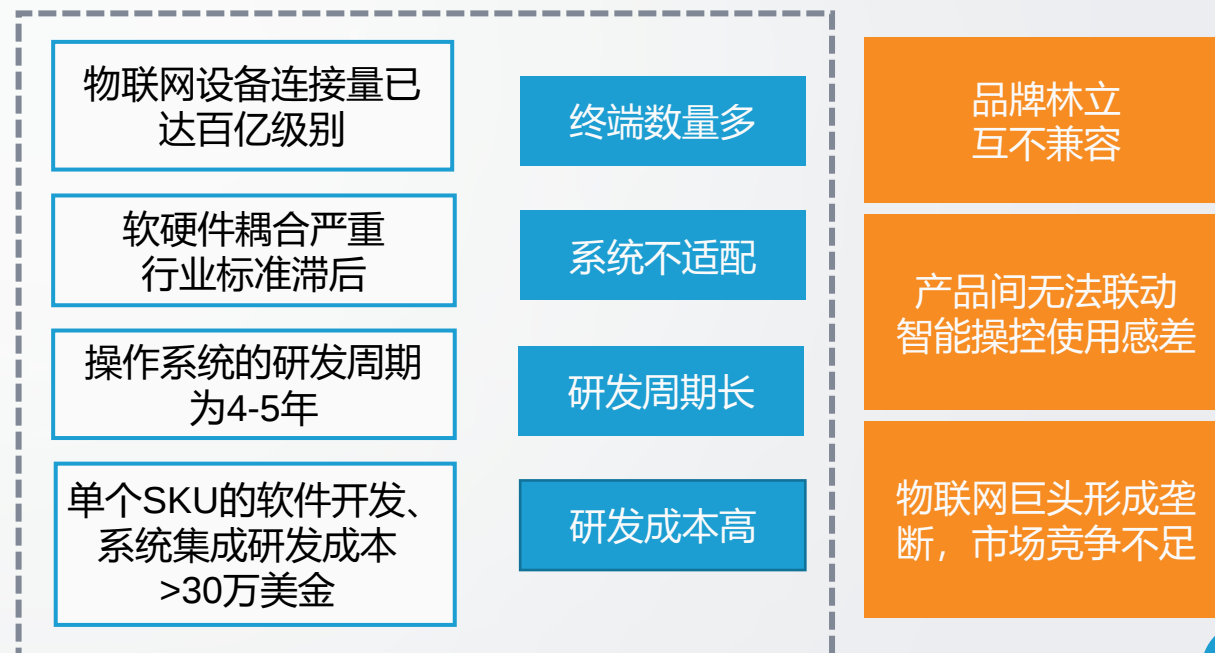
我们发现物联网发展不及预期重要原因之一是上层软件应用开发和底层硬件紧密耦合，很难由第三方开发者为硬件开发应用。这导致物联网软件应用迭代慢、成本高，生态闭塞。

亿欧智库：2014-2020年物联网接入设备的预期和实际差距



数据来源：预测数据来自Gartner、IBM、思科等预期综合；实际数据来自Gartner2019。

亿欧智库：操作系统是制约物联网发展的重要因素



# 区别于传统操作系统，物联网在操作系统领域需要易拓展、跨平台、低成本、强安全、稳连接的新要求

物联网的终端设备数量呈现指数型上涨，物联网终端软件复杂度的提升势在必行。物联网开发平台也需要具备一套对软件开发者更友好的、相对统一的操作系统，使软件开发过程从硬件开发中解耦出来。只需要基于操作系统，即可快速开发上层应用。

## 亿欧智库：物联网发展对操作系统提出的新要求



亿欧

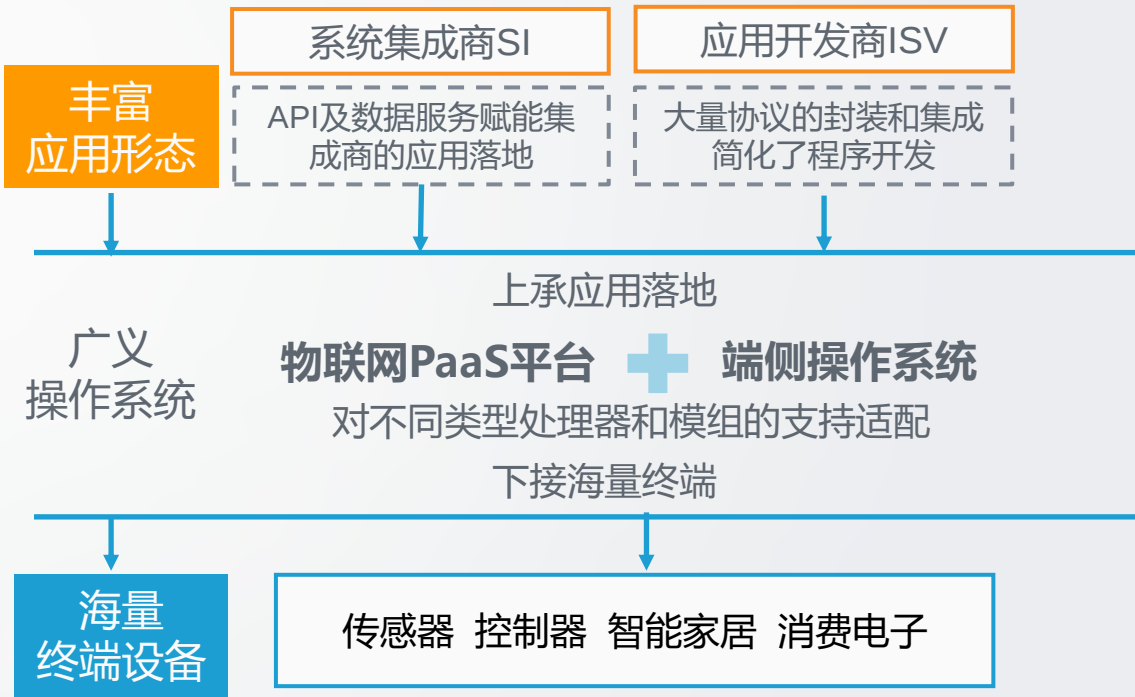
广义物联网操作系统上承落地应用，下接海量终端

物联网产业如果按照功能进行分层，可以划分为感知层、网络层、平台层和应用层。感知层相当于物联网系统的五官和四肢，负责前端数据的收集；网络层相当于神经纤维，用于高速传递数据和信息；平台层相当于小脑和大脑，连接管理平台确保物联网系统自主运行，而操作系统则确保智能应用有运行的基底。**广义的操作系统是云端PaaS平台和端侧系统的统一和融合**。从市场分额占比来看，平台层占有物联网大约20%市场。**综合技术因素和市场规模，操作系统是物联网发展的重要一环。**

亿欧智库：物联网产业按功能进行分层



亿欧智库：广义操作系统起到承上启下的作用

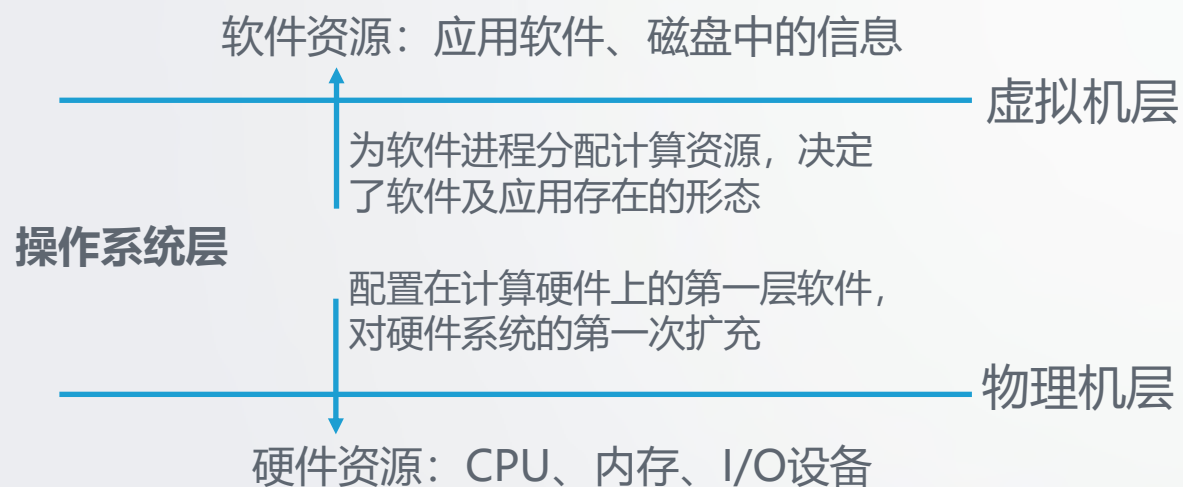


# 狭义操作系统协同软硬件资源，提高终端效率

操作系统是硬件资源和软件应用之间的桥梁，对于复杂的底层硬件资源而言，软件开发必须通过操作系统的指令和接口进行调用。通过操作系统可实现以下几方面的功能。

- ◆ 尽量有效、合理地组织和管理计算机的各种软硬件资源；
- ◆ 合理地组织计算机的工作进程，控制程序的执行并在程序执行的过程中向程序提供各种各样的服务功能；
- ◆ 用各种各样的方式使得用户能够非常灵活、方便地使用计算机，采用各种各样的技术使整个计算机系统能够高效率地运行。

## 亿欧智库：操作系统是软硬件资源的分配者



## 亿欧智库：操作系统的五大管理功能



# 亿欧 终端应用碎片化，物联网操作系统市场需求大

碎片化是物联网终端发展的最重要特征之一。从本质上来讲，有多少种终端设备就应该有多少种上层应用。万物互联时代不仅仅是物与物之间的实时通讯，也包含了相当部分的人与物无缝互动。云侧和端侧的功能需要去同时实现物与物实时通讯及人与物的无缝互动。我们判断，在智慧城市、工业应用等场景中，云端平台的赋能将发挥更大的作用，而在室内智能和消费电子场景，人与物的无缝互动是未来重点关注的方向。**操作系统在人与物的无缝互动中将发挥最重要的作用，发展潜力和市场需求巨大。本报告核心关注于端侧操作系统的现状、发展和未来。**

向右越来越依靠终端操作系统赋予的交互和功能



向左越来越依靠云端平台赋予的设备联网管理及数据分析自动化能力

# 全球科技巨头正在以“云+端”模式布局物联网

我们观察到，近年来世界范围内的科技巨头都开始在物联网领域进行重点布局，虽然各家打法各有差异，但**总结来看巨头布局互联网主要关注“云+端”模式**，努力构建生态和竞争优势。在**云侧**，以云平台为支撑的中间件服务、数据服务和信息服务，以及在**端侧**，以底层软硬件设备为基础的配套组件，将会是巨头控制物联网海量终端、丰富应用形态的两个最好**切入点**。

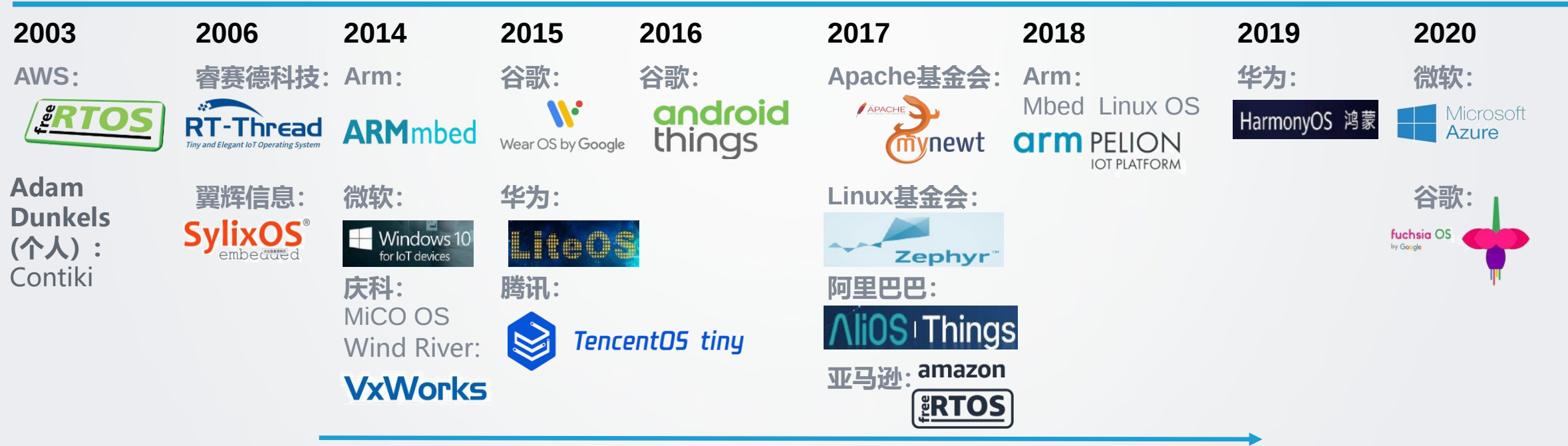
亿欧智库：国际范围内领先的科技企业在物联网方向的布局

| 代表公司 | 亚马逊                                                            | 苹果                                                                                   | 微软                                                                                                               | 华为                                                                     | 阿里巴巴                                                                   |
|------|----------------------------------------------------------------|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|------------------------------------------------------------------------|
| 云    | 依托AWS的云服务：<br>AWS IoT<br>集成人工智能和大数<br>据分析能力                    | 针对硬件产品及场景的开<br>放平台：<br>Homekit、HealthKit、<br>SiriKit等<br>助力开发者应用开发                   | Azure 物联网中心：<br>受信任的平台模块 (TPM)、<br>Azure IoT Hub、Azure<br>Stream Analytics、Azure<br>Storage、Azure<br>DocumentDB等 | 华为云，物联网云平台：<br>OceanConnect                                            | 阿里云物联网平台：<br>开发者服务工具IoT Studio<br>物联网市场Link Market<br>垂直行业平台：<br>飞凤、飞燕 |
| 端    | 终端AI芯片：<br>AWS Inferentia<br>提升AI语音助手效率；<br>操作系统：<br>Free RTOS | 芯片：<br>针对硬件产品线推出<br>A/T/H/S系列芯片<br>操作系统：<br>与设备适配的iOS<br>传感器技术：NFC、3D<br>传感器、生物信息传感器 | IoT芯片：<br>Azure Sphere系列认证芯片<br>操作系统：<br>Windows 10 IoT<br>(包含核心板、企业版)                                           | IoT芯片：<br>NB-IoT商用芯片<br>IoT通信模组：<br>eLTE-IoT模组<br>操作系统：<br>LiteOS、鸿蒙OS | 通信芯片：LoRa芯片<br>AI芯片：平头哥RISC-V架构<br>操作系统：AliOS Things                   |

# 物联网操作系统是科技巨头的重要着力点，也是取得物联网时代战略制高点的关键之一

物联网操作系统尚处于发展早期阶段，呈现出碎片化状态。目前市场中存在着**不同来源的物联网操作系统**：有从原有嵌入式操作系统改进而来的，也有从通用系统裁剪而来的，还有面向物联网需要完全从零发展而来的。不同行业背景的厂商（操作系统大厂、硬件厂商、云计算服务厂商等）从自身业务需求出发进行探索。我们判断尽管目前物联网市场中操作系统的市场占有率较为分散，但未来有可能出现几个垄断具体业务场景或生态的主导型物联网操作系统。

亿欧智库：主流的物联网操作系统发展历史



物联网操作系统从服务连接，到服务应用，最终目的应该是服务于智能

来源：亿欧智库整理

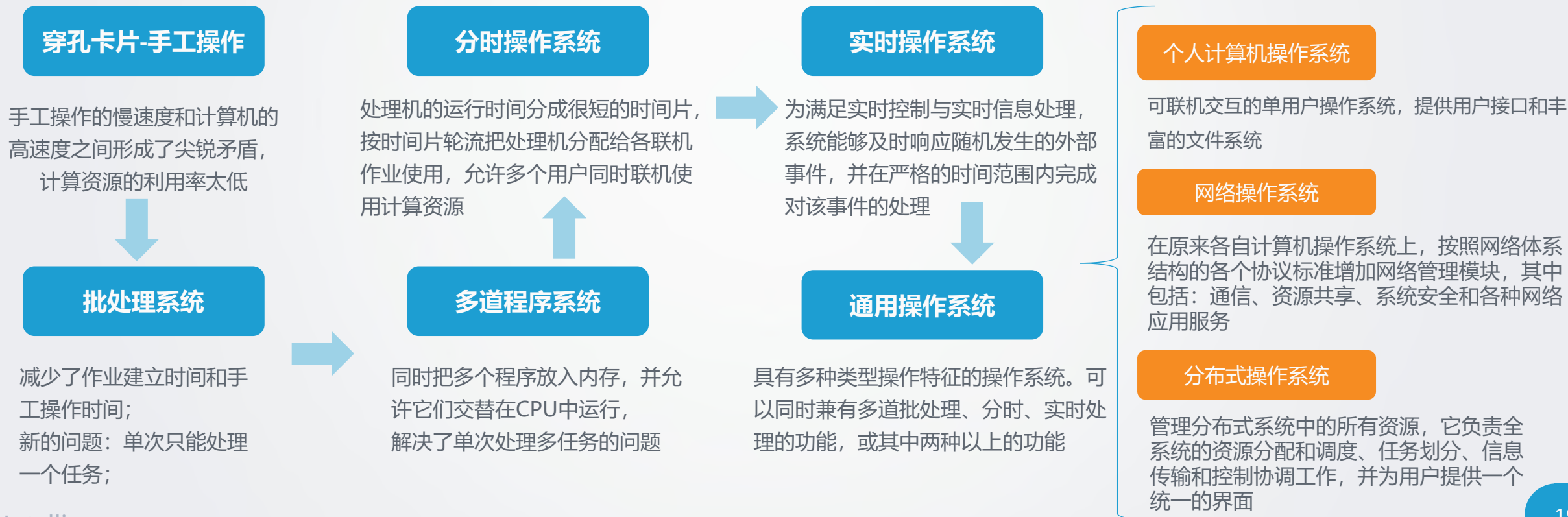
## 二、操作系统历史经验回顾

Operating system historical experience review

# 亿欧 操作系统的出现和迭代是时代需求和技术制约下的平衡

操作系统管理硬件资源，控制程序运行，改善人机界面，和为应用程序提供支持的系统软件。是为满足用户使用计算机过程中提高资源利用效率、增强计算机系统性能的产物。在不同时代的需求和技术发展与制约下，操作系统呈现出不同形态，可以确认的是，其会一直发在时代需求和技术制约间寻找平衡。

## 亿欧智库：操作系统从简单到复杂，出现面向特定场景的复杂系统



# 操作系统不仅带来技术上的进步，同时也带动商业模式的创新

操作系统OS自诞生之初，就是硬件设备向更复杂阶段发展必然产物。早期硬件资源少，产品功能简单，一体裸机也可以实现用户需求。而随着硬件复杂度的提升，和对硬件资源利用效率的提升，操作系统逐渐成为硬件和使用者之间的必需品。实时操作系统RTOS的出现是为了更容易实现实时多任务编程，标准化通用操作系统的出现则是为了多用户动态装载应用。历史经验告诉我们，操作系统不仅仅带来技术上的进步，同时也是产业分工和商业模式的演化，不单单是技术问题。

## 亿欧智库：操作系统发展带来技术和商业模式上的进步

### 操作系统发展带来的技术上进步

- 复用代码，减少软件开发者的重复劳动；
- 功能模块化，便于专业分工，利于协同开发；
- 屏蔽掉无关细节，降低软件开发难度；
- 标准化接口，高内聚，低耦合。

### 操作系统发展带来的商业模式上的创新

- 产业上下游之间进行业务区分，便于业务分离和专业化；
- 不断激发产品形态和业务模式的创新；
- 成功的操作系统一直在帮助伟大企业构建竞争壁垒。

操作系统的发展史上出现过不少成功案例，也出现过更多失败的案例，即使是行业巨头也难免曾经折戟。我们相信未来成功案例和失败案例集上的名单会不断增加。

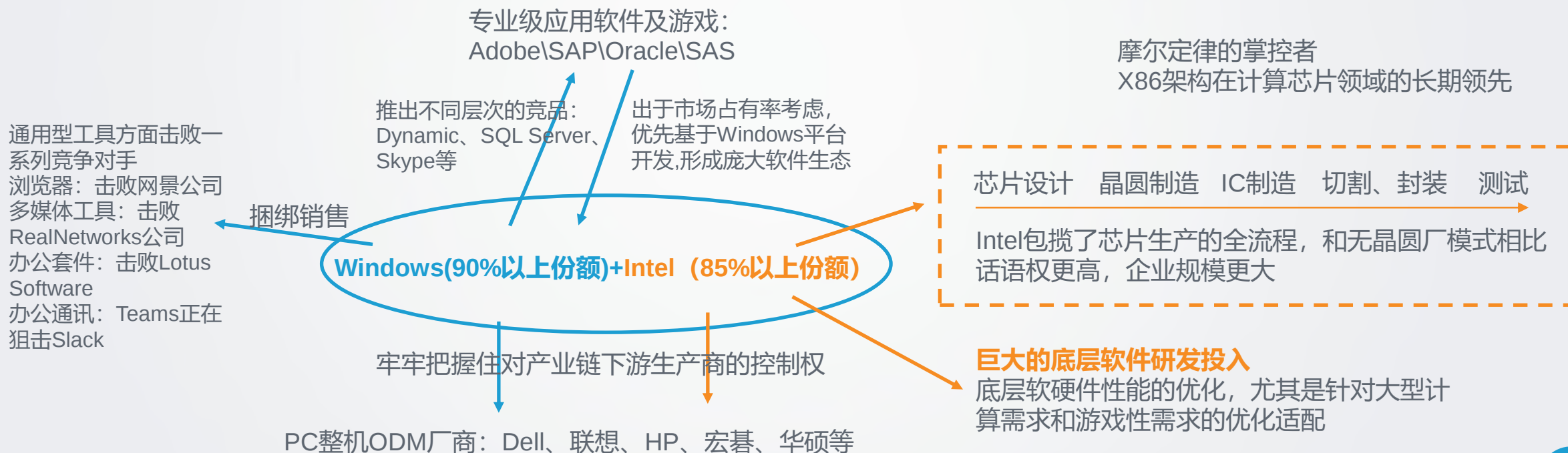
## 亿欧智库：历史上以失败告终的操作系统案例

- 1964年由贝尔实验室、麻省理工大学和通用电气共同研发的大型主机操作系统**Multics**，受限于当时的技术条件，研发设计目标难度太大，项目进展缓慢，于1969年被撤项；
- 1991年，IBM投入20亿美元打造**WorkSpace**操作系统，由于目标设置过于宏大复杂，在历时4年开发后以失败告终；
- 1994年，苹果为追赶微软windows操作系统而设立的代号为**Copland**的项目，最终因为该PC系统无法稳定运行而以失败告终；
- 2007年，微软推出的**Windows Vista版**因为无法对当时的硬件发展水平有很好的兼容而遇冷。
- 2015年，微软持续经营10余年的移动端操作系统**Windows Phone**在与安卓的竞争中宣告失败。

# PC桌面时代的垄断者：Wintel联盟

Windows操作系统与Intel CPU组成的个人计算机架构，称霸了桌面端20余年。英特尔沿着摩尔定律不断发展以提升硬件水平，微软Windows系统升级将升级的硬件水平消耗掉，这又被总结为安迪-比尔定律。软件和硬件之间的协同进步关系使得Wintel双方在各自领域都获得了竞争优势，如下图所示，微软在软件市场中形成对应用软件开发商的绝对优势，Intel则成为整个半导体产业中技术实力最强的企业。Wintel联盟为个人电脑的推广和使用做出了重大贡献，但也存在不同程度的排他性。

## 亿欧智库：PC时代Wintel联盟生态体系



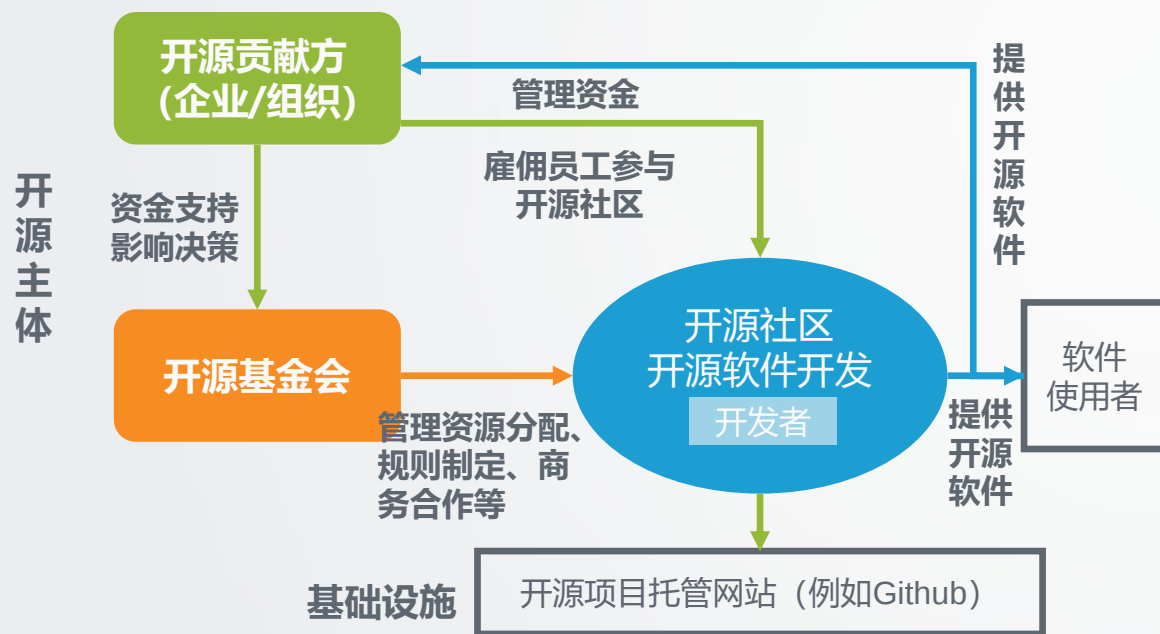
各自领域的单一垄断厂商构成**安迪-比尔定律**

来源：亿欧智库整理

# 亿欧 破除商业公司对操作系统的垄断：Linux的开源发展模式

**Linux操作系统**起源于1991年，从创始之初就采用内核开源的模式吸引个人和开发团体参与到开源社区的代码贡献中。到目前为止，Linux已经形成成熟的开源社区，汇聚了一大批开发者，核心代码量达到2500万行以上，社区版本和商业发行版本广泛应用于服务器、嵌入式和开发者的桌面端。Linux模式很好地印证了以**开源基金会**、**开源社区**和**开源贡献方**为主要参与方的模式也可以为全社会贡献高性能的底层软件。

## 亿欧智库：开源软件的组织模式



来源：中国信通院

GNU是“GNUs Not Unix”的缩写，为Linux发行版本开发提供了大量的自由软件。所以Linux 内核+GNU 项目构成了完整的GNU/Linux 发行版。

Linux发行版本通过通用公共许可证 (GPL) 发布。软件在GPL协议下，自由软件都遵循非版权原则，允许用户自由拷贝、修改和销售，但源代码的修改必须向所有用户公开。

## 亿欧智库：GNU+Linux内核构成Linux发行版

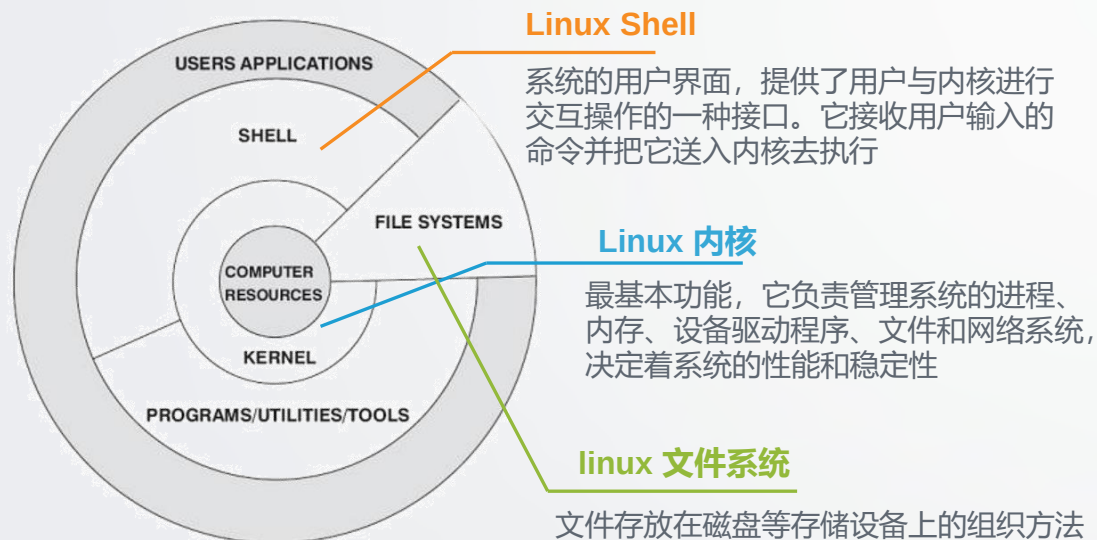


来源：东方证券

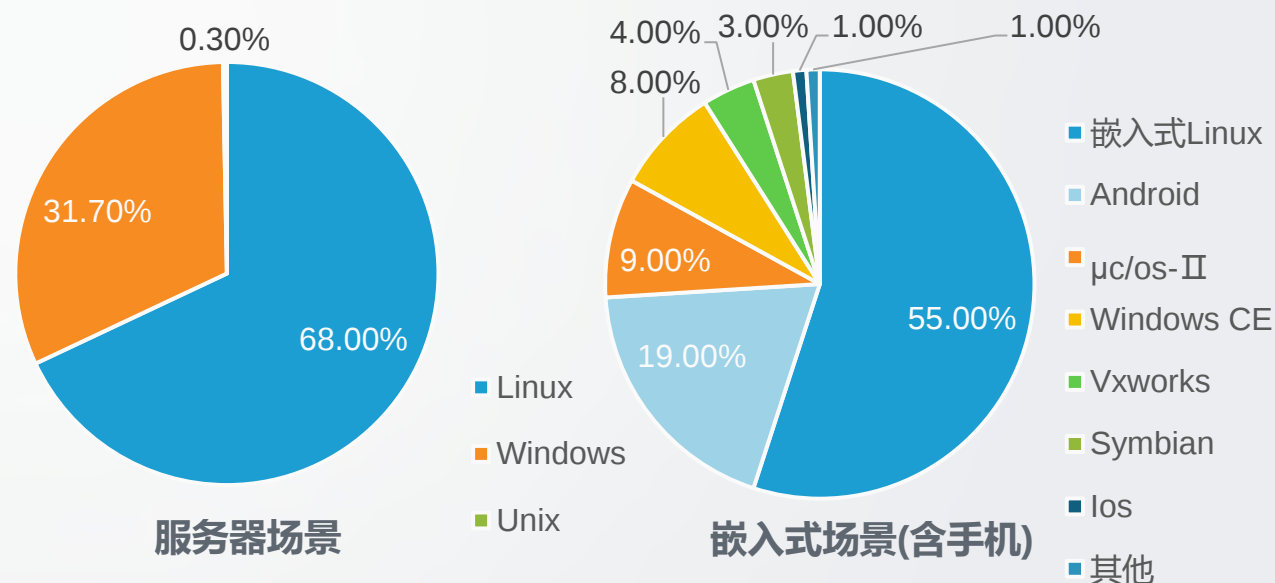
# Linux的开放性使其在开发者场景（服务器+嵌入式）广泛应用

Linux主要由**内核、shell、文件系统和应用程序四部分组成**。内核、shell和文件系统一起形成了基本的**操作系统结构**，它们使得用户可以运行程序、管理文件并使用系统。正是受益于Linux内核的开放性和自定义程度高，相比 Windows 等闭源系统，开发者使用计算机资源时具有更高自由度，因此在服务器及嵌入式等开发者使用居多的场景下拥有很高的市场占有率。而根据IDC于2017年公布的数据，全球服务器操作系统（免费+付费）中有约68%采用的是Linux服务器操作系统。而根据2016年的市场调研结果，嵌入式Linux和基于Linux的Android占有嵌入式系统市场的74%。

亿欧智库：Linux的内核、shell和文件系统



亿欧智库：Linux的服务器及嵌入式市场占有率



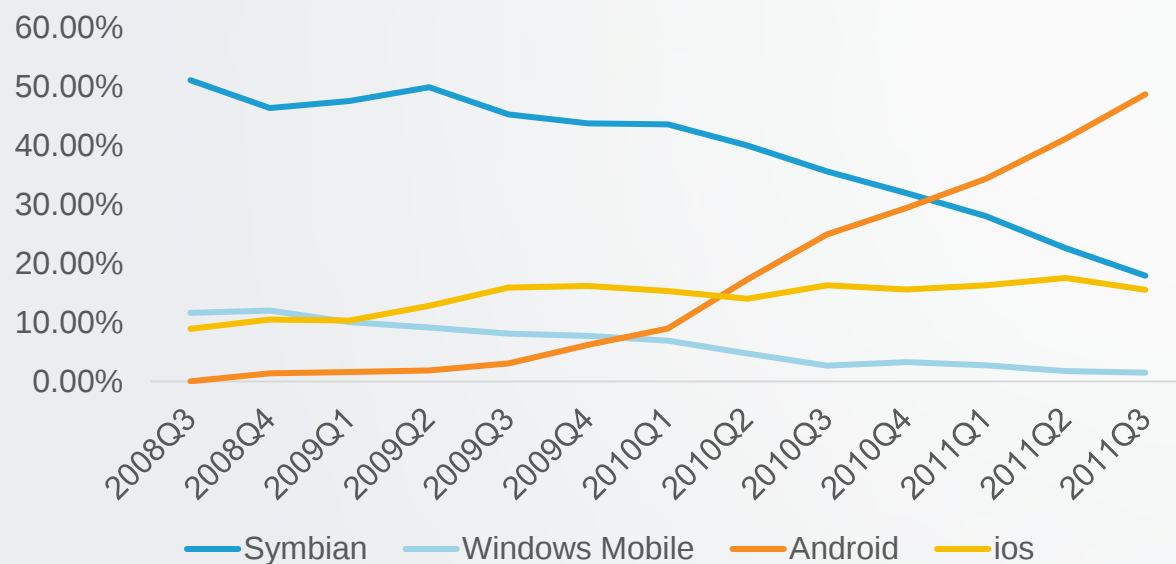
来源：IDC,2017

来源：21ic,2016

# Windows Phone：发展延续长达20年的移动端设备操作系统 终失败

Windows Phone的历史最早可追溯到1996年研发的PDA系统WinCE，桌面操作系统赢家微软早已开始布局移动端——能够随身携带的袖珍型个人计算机。经过10余年的发展，由微软开发的移动端操作系统Windows Mobile 6还是取得了不错的成绩，在2007年的手机市场中占有30%份额，主要和Palm、黑莓、Symbian这三者展开竞争。在iOS和Android出现之前，Windows Phone是市场中最早支持触屏的系统。而当iOS和Android开始迅速取代传统手机的市场时，微软却开始逐渐败退。

亿欧智库：2008Q4季度-2011Q3季度智能手机操作系统市场份额



2008-2011年间，移动端OS的交替阶段，Windows Mobile明显落后于Android和iOS

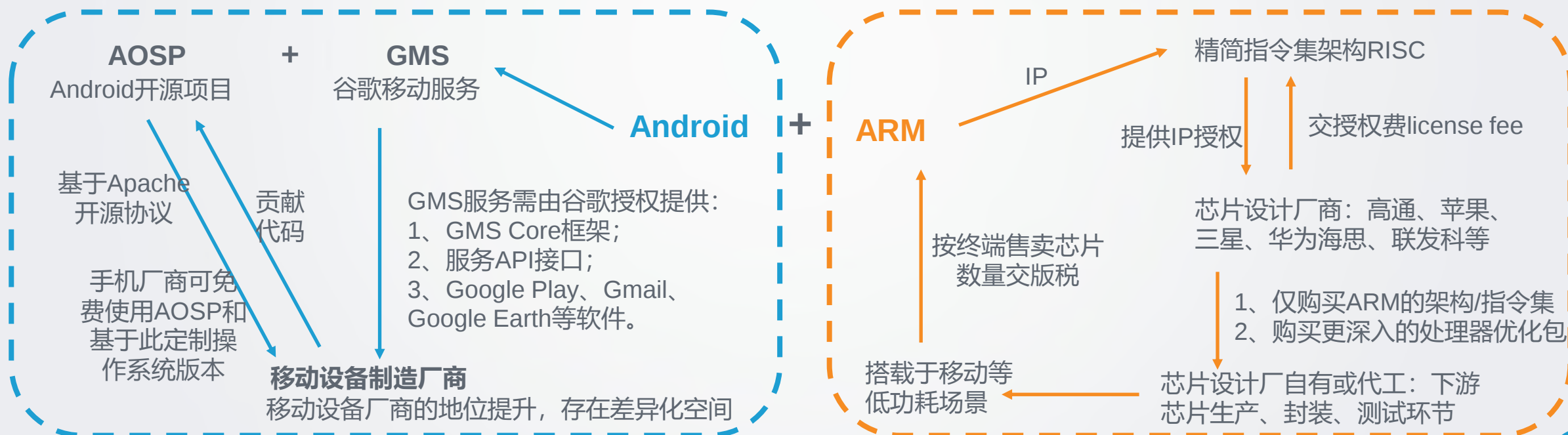
亿欧智库：相比于Android，Windows Phone犯的错误

|        | Windows Phone                                                                                                                              | Android                                                                                                                        |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| 对硬件厂商  | <ul style="list-style-type: none"><li>● 每台手机向制造商收取10-20美元的Windows Phone系统授权费用</li><li>● 手机制造商无法开发低价机</li><li>● 所有机型的UI相同，OEM思维严重</li></ul> | <ul style="list-style-type: none"><li>● 系统向手机制造商免费授权，内核代码开源</li><li>● 手机制造商据此可以采取差异化、低价化</li><li>● 硬件厂商可定制自有的上层系统和UI</li></ul> |
| 对软件开发者 | <ul style="list-style-type: none"><li>● 新版本推出速度慢</li><li>● 出现过重大内核和开发框架更替，开发者技能断层</li></ul>                                                | <ul style="list-style-type: none"><li>● 新版本推出迭代快速</li><li>● 未出现过开发者技能断层</li></ul>                                              |
| 最终对消费者 | <ul style="list-style-type: none"><li>● 软硬件水平相对于价格不存在竞争力</li></ul>                                                                         | <ul style="list-style-type: none"><li>● 在中低端机型上占据竞争力</li></ul>                                                                 |

# 基于Linux内核，Android+ARM模式在移动端取得成功

尽管Linux系统在桌面端远远落后于Windows，但其开源模式却为其他系统生态提供机会。以Linux为内核的安卓系统在移动端大放异彩。由ARM公司主导的硬件开放和由谷歌公司主导的软件开源模式是移动时代Android+ARM联盟建立的成因。相比于联系紧密的Wintel联盟，Android+ARM的结构要松散的多。Android由开源项目AOSP（Android Open-Source Project）和谷歌提供的GMS（Google Mobile Service）服务组成。ARM作为芯片IP授权方，带动了中下游fabless和foundry厂商的繁荣。

## 亿欧智库：移动时代Android+ARM联盟以更开放的形式拓展市场



 亿欧

# AOSP+GMS：谷歌拿下手机端海量市场的重要组合

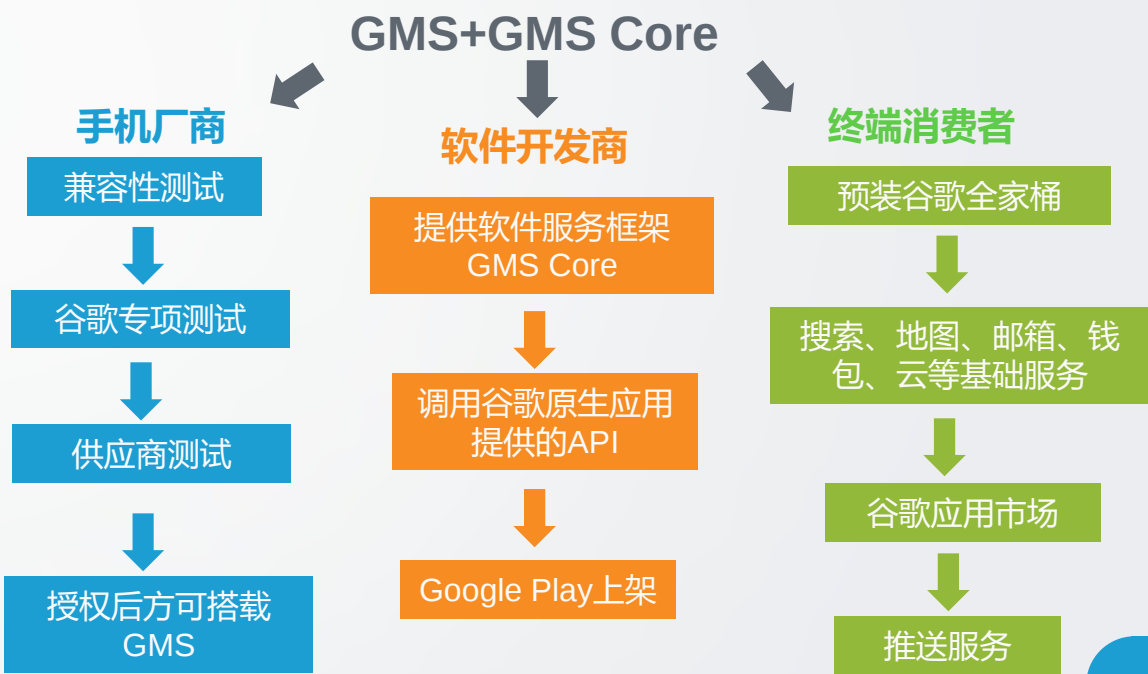
移动端操作系统经历过相当长一段时间的争夺，谷歌Android、苹果iOS和微软WinPhone都曾是有有力竞争对手。我们认为相比于WinPhone和iOS的封闭性，正是Android系统将核心的软件本体以开源形式提供给手机厂商使用才得到众多厂商的支持，厂家可以在开源系统的基础上进行自由开发，从而构建起与其他厂商的差异性。而GMS部分使得谷歌对开源部分具有控制权，强化其对软件的监管。AOSP+GMS的开放与控制互补是谷歌Android能占据移动端80%以上设备的根本原因。

亿欧智库： AOSP及Android开放手机联盟（Open Handset Alliance）

| 企业分类     | 企业名单（部分）                                                                          |
|----------|-----------------------------------------------------------------------------------|
| 通信运营商    | 中国电信、中国移动、中国联通、KDDI、NTT DoCoMo、T-Mobile、Sprint Nextel                             |
| 半导体公司    | ARM、Broadcom、Intel、MediaTek、Marvell、NVIDIA、Qualcomm、ST-Ericsson、Texas Instruments |
| 手机及其他终端厂 | Acer、ASUS、Dell、FIH、Kyocera、LG、Motorola、Samsung、Sharp                              |
| 软件供应商    | ACCESS、Google、MOTOYA、NXP Software、SkyPop、SVOX、Esmertec                            |
| 其他商业公司   | Accenture、Aplix、Teleca AB                                                         |

来源：网络

亿欧智库： GMS作为谷歌控制整个生态圈的钥匙



操作系统作为所有软件应用的根本，因其技术含量之高而成为ICT行业金字塔尖的存在。操作系统同时离不开硬件厂商和软件开发者的支持，成功的操作系统一定具有丰富的软件生态支撑。通过对前两次重大操作系统更替的回顾，我们总结了以下要点：

- 1、操作系统永远满足的是时代的需求，计算机系统曾经发生过的更替还会继续发生。**在底层技术和市场需求发生变化时，总有厂商可以敏锐地发现并把握住时代更替的机会，以新技术+新模式替换掉旧有系统。
- 2、在相同类型终端上的追赶几乎不可能实现。**据统计，Win10操作系统在2018年已经具有3500万应用，超1.75亿个软件版本。支持1600万个硬件\驱动组合，而截至2017年底，Google Play共有APP 350万个，App Store共有APP 210万个。追赶者很难通过对开发者的激励来形成一个新的软件生态，这么做的收益和付出处于不对称的状态。
- 3、从0到1是操作系统弯道超车的最好时机，旧有经验可能成为阻碍。**微软和谷歌在推出移动端操作系统的时间节点上相差不多，在2011年前两者的市占率一直处于焦灼状态，而最终Android凭借着更开放的软硬件生态赢得了开发者的青睐，WinPhone系统消失在历史中。
- 4、更为开放的Android+ARM模式取代了封闭的Wintel模式，**更多的企业可以从开放产业链中获得利益，硬件设备和上层应用也更为丰富。在碎片化的物联网终端上，我们认为开放模式带给整个产业链的机会和促进会更大，尤其是在物联网产业内。

当然，在传统PC时代和移动时代，较少出现应用上的异构计算和不同类型操作系统混合作业的形式，因此物联网的系统发展会更加复杂。

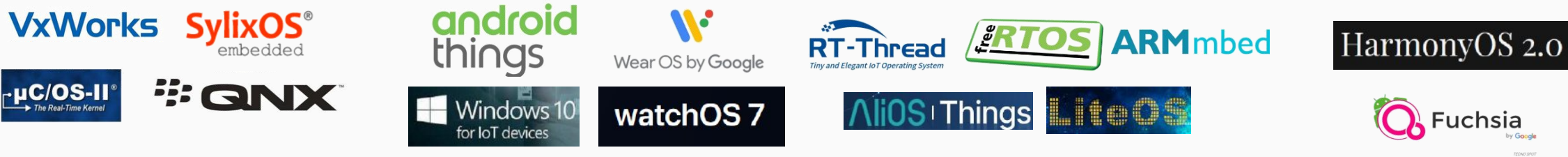
# 三、物联网操作系统代表类型及案例

Representative types and cases of IoT OS

# 四大类物联网操作系统，起源和发展方向各有不同

物联网操作系统尚处于发展早期阶段。**整个物联网操作系统市场呈现出碎片化状态。**目前市场中至少存在着四种发展来源的物联网操作系统：**嵌入式实时系统及升级版、原有通用系统裁剪定制后适应部分物联网需求，近几年来出现的专门为物联网设计的轻量级IoT OS，以及完全从新开发的新型操作系统。**不同背景的厂商从自身需求和实际业务能力出发进行探索。我们判断尽管目前物联网市场中操作系统的市场占有率较为分散，但未来有可能出现几个垄断应用场景的物联网操作系统。

亿欧智库：基于现有市场的四类物联网操作系统



1、传统嵌入式系统+通信协议+其他物联网功能模块  
代表系统：SylixOS、µC/OS-II、Vxworks等  
优点：更偏向硬件层，硬件推广成本低；  
缺点：软件开发难度高，软硬件耦合较严重，生态封闭。

2、基于Linux、iOS、Windows等成熟操作系统的裁剪  
代表系统：uclinux、win10 IOT、WatchOS、Android Things等  
优点：软件开发工具完善，开发者基数大；  
缺点：原版系统冗余严重、历史遗留问题较多，普遍不适应低功耗场景、可扩展性低，需要定制化。

3、面向物联网的轻量级IoT OS  
代表系统：RT-Thread、华为 LiteOS、ARM mbedOS、AliOS Things、Azure RTOS、Amazon FreeRTOS等  
优点：专门为物联网开发设计，联网协议完善，开发包齐全；  
缺点：仍未解决物联网终端碎片化和割裂的问题。

4、下一代操作系统  
代表系统：谷歌Fuchsia、华为鸿蒙OS  
优点：是对嵌入式系统和个人设备操作系统的革命性进化，理念为统一型OS，扩展性及移植性好；  
缺点：开发面临的不确定性高，缺少开发者基础和生态支持。

来源：亿欧智库整理

 亿欧

# 各类物联网操作系统的应用场景有所差异

物联网操作系统按照应用场景来看，各类操作系统都有其对应的重点应用场景，而在部分场景中存在有多类操作系统可供选择的情况。

亿欧智库：现有四类物联网操作系统的应用场景（举例）

|          | 操作系统名称             | 应用场景                                                                                      |
|----------|--------------------|-------------------------------------------------------------------------------------------|
| 传统嵌入式系统  | uClinux            | 工业控制、无人机、因特网设备(路由器、交换机、防火墙、负载均衡器等)、专用的控制系统(自动售货机)                                         |
|          | Vxworks            | 通信、军事、航空航天、工业制造机器人、医疗器械、及汽车电子                                                             |
|          | SylixOS            | 航空航天、电力电网、轨道交通、机器人、新能源、医疗、工业自动化                                                           |
| 通用OS裁定制后 | Android things     | 应用于除手机（Android）、电视（Android TV）、车机（Android Auto）和穿戴式设备（Android Wear）之外的物联网及嵌入式设备，适用于构建智能家居 |
|          | Windows 10 IoT 核心板 | 智能建筑、IoT 网关、HMI、智能家居、可穿戴设备                                                                |
|          | Windows 10 IoT 企业版 | 工业平板电脑、零售服务点、ATM、医疗设备、制造设备、瘦客户端                                                           |

|            | 操作系统名称          | 应用场景                                           |
|------------|-----------------|------------------------------------------------|
| 轻量级物联网操作系统 | RT-Thread       | 智能家电、网关、工业物联终端                                 |
|            | Alios things    | 智能生活电器、智能穿戴设备、工业数据采集设备                         |
|            | Huawei LiteOS   | 智能家居、摄像头、智能水表、智能停车、智能照明                        |
|            | Amazon FreeRTOS | 使用微控制器，RAM和ROM资源较少的应用场景，如工业场景下的传感器、制动器、泵和自动化组件 |
| 统一操作系统     | HarmonyOS 鸿蒙    | 依次推进从低资源硬件到高资源硬件的覆盖，以分布式能力面向全场景                |
|            | Fuchsia         | 可以兼容手机、PC、智能家居等物联网场景的统一操作系统                    |

来源：亿欧智库整理

---

# 嵌入式操作系统

Embedded Operating System

---

# 嵌入式系统逐渐取代裸机系统，物联网操作系统发展开启早期阶段

早期嵌入式开发中并没有操作系统的概念，而是直接在裸机上写简单的轮询系统或前后台系统。但裸机系统很难保证程序运行的实时性，因此随着硬件水平的提升，MCU的Flash和RAM越来越满足操作系统的占用，所以由嵌入式系统带来的开发模块化工具和实时响应的优势反而简化了在中大型项目开发中的难度，因此嵌入式实时系统逐渐取代裸机系统成为趋势。

亿欧智库：几种任务处理系统的对比

| 模型    | 事件响应 | 事件处理 | 特点            | 优缺点                              |
|-------|------|------|---------------|----------------------------------|
| 轮询系统  | 主程序  | 主程序  | 轮询响应事件，轮询处理事件 | 只适合顺序执行的功能代码，当有外部事件驱动时，实时性就会降低   |
| 前后台系统 | 中断   | 主程序  | 实时响应事件，轮询处理事件 | 尽管将事件的响应和处理分开，但事件的处理还是在后台内按顺序执行的 |
| 多线程系统 | 中断   | 线程   | 实时响应事件，实时处理事件 | 优先级概念，相比前后台系统，多线程系统的实时性又提升       |

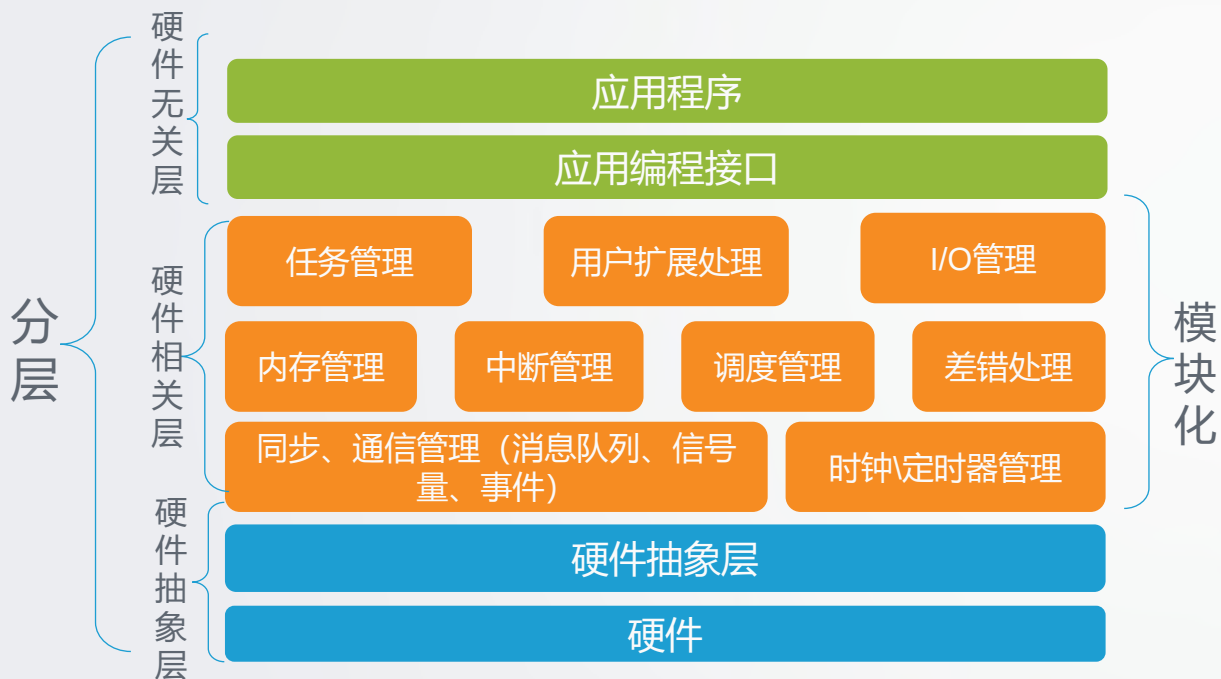
亿欧智库：前后台系统和嵌入式实时操作系统的对比

|           | 前后台系统                                               | 实时操作系统                                        |
|-----------|-----------------------------------------------------|-----------------------------------------------|
| 资源占用      | 不需要分配资源给OS                                          | 需要分配资源给OS（不同系统内核资源使用情况不同），内核资源也不同，可通过配置资源进行配置 |
| 开发难度      | 不需要学习OS的API，上手简单；但后续开发难度提升                          | 开发者需要熟悉OS的基本操作（任务建立、操作，任务间通信、优先级处理、中断处理）      |
| 实时性       | 难以确保每个操作能够实时响应，如果一个函数花费过长时间，将使整个系统的实时性下降。实时响应，轮询处理。 | 实时嵌入式OS的调度算法可以最大程度保证系统的实时性；实时响应、实时处理。         |
| 可拓展性、可维护性 | 可拓展性差，可维护性差                                         | 模块化、结构清晰                                      |

# 嵌入式操作系统分层模块化的结构决定了其可移植、可扩展的特征，微内核的结构决定了其可靠灵活的特征

结构是操作系统的基础，它定义了硬件与软件的界限、内核与操作系统其它组件（文件、网络、GUI等）的组织关系、系统与应用的接口。结构是确保系统可靠性、灵活性、可移植性、可扩展性的关键。目前嵌入式操作系统主要采用**分层和模块化相结合的结构**以及**微内核结构**。

亿欧智库：嵌入式系统的分层模块化模式



分层模块化带给嵌入式系统优势在于，移植工作集中于模块化的硬件相关层，系统的可移植性和可伸缩性都得到保证。

亿欧智库：嵌入式系统的微内核结构



微内核的优势在于，可以向外提供一致接口，以及分布式系统支持。但是，通过微内核发送信息、接受应答并解码所花费的时间比进行一次系统调用的时间多。

# 嵌入式操作系统的实时性提高系统性能、可移植性有效降低研发成本，为物联网的系统开发提供了良好基础

**嵌入式系统是物联网开发的基础。**嵌入式操作系统最初应用于单片机，以可编程控制器的形式出现。从无操作系统支持的裸机，发展到高度模块化和拓展性的实时操作系统（RTOS）。文件和目录管理、设备管理、多任务、网络、图形用户界面（GUI）等功能，并提供了大量的应用程序接口（API），使得软件开发的难度不断下降，自由度不断上升。作为软硬件资源的控制中心，以尽量合理有效的方式解决多用户共享下的资源。大部分的嵌入式系统都具有软实时性，可移植于多种处理器，并且基于优先级设计。总结而言，嵌入式操作系统具有多种优点，尤其适用于成本较低、计算资源不多、对实时性有要求的场景。

亿欧智库：嵌入式操作系统发展的四个阶段及其特点

| 无操作系统的嵌入式算法    | 监控式实时操作系统     | 通用嵌入式实时操作系统           | 云端连接的操作系统        |
|----------------|---------------|-----------------------|------------------|
| 4-8位单片机为主      | 8-16位嵌入式处理器为主 | 32位RISC微处理器           | 32位及以上微处理器       |
| 无操作系统支持的裸机     | 以简单监控式操作系统为主  | 代表：Vxworks、Windows CE | 嵌入式系统和网络的连接      |
| 汇编语言对系统进行直接控制  | 处理器种类繁多，通用性差  | 可运行在不同的微处理器上          | 可运行在不同的微处理器上     |
| 系统结构及功能单一，针对性强 | 用户界面不够友好      | 具有丰富API和嵌入式应用软件       | 具有丰富API接口，以及云端能力 |

亿欧智库：嵌入式操作系统相对于其他系统具备的特征



亿欧

# 嵌入式软件开发的特性使得开发工具套件也相当重要

虽然嵌入式软件的代码规模较小，但开发难度大。首先，嵌入式软件的开发需由交叉编译的方式完成，负责开发的宿主机平台和负责运行的目标机之间需通过物理连接或逻辑连接。其次，难度主要集中于其既涉及底层软件的开发，又要基于操作系统做上层开发，极端条件下甚至要直接开发一个新的操作系统。并且嵌入式开发调试的工具远没有一般的软件开发工具方便易用，需要具有较多的软硬件开发经验。因此，辅助软件开发的工具软件也具有重要分量，主要包括交叉开发工具、仿真开发工具、测试工具、配置管理工具和维护工具等。

亿欧智库：嵌入式软件开发需要在宿主机和目标机之间交叉开发



亿欧智库：嵌入式软件开发工具的分类



# 嵌入式操作系统的典型代表——μC/OS-II

μC/OS-II是一个抢占式实时多任务内核。它是用ANSI标准的C语言编写的，包含一小部分汇编语言代码，使之可以提供给不同架构的微处理器使用。μC/OS应用领域包括：航空业、医疗器械、网络设备以及工业机器人等。μC/OS-II 具有可移植、可固化和可裁剪的优点。μC/OS-II内核的主要功能是任务的调度和切换，采用抢占式调度策略，保证任务的实时性，能够管理多达64个任务。

## 亿欧智库： μC/OS-II的核心特征

### ● 可移植性 (Portable)

μC/OS-II的源码用移植性很强的ANSI C，和微处理器硬件相关的部分用汇编语言。汇编语言写的部分已经压到最低限度，使得μC/OS-II便于移植到大多数8位、16位、32位以至64位微处理器、微控制器、数字信号处理器（DSP）上。

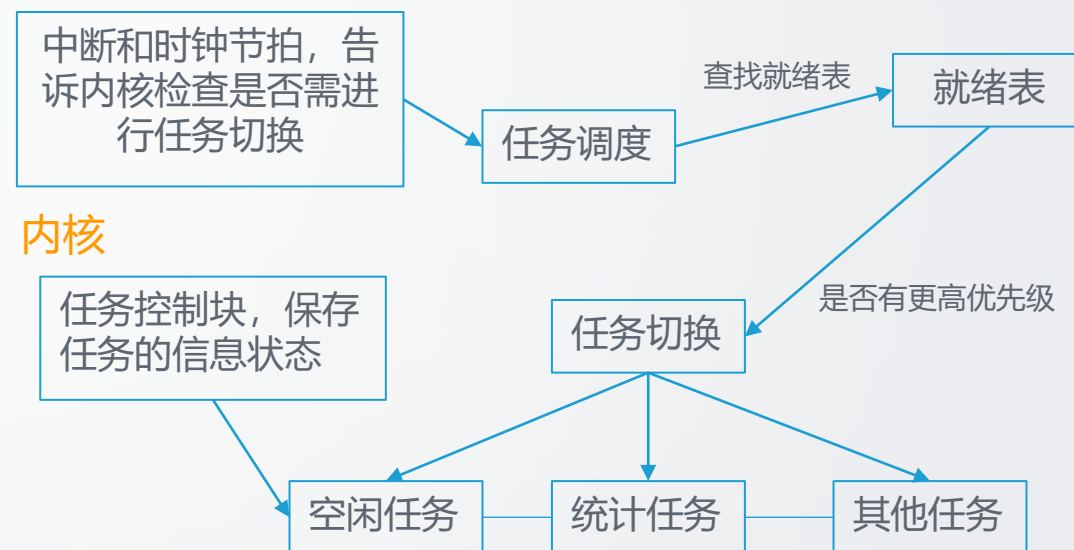
### ● 可固化 (ROMable)

μC/OS-II为嵌入式应用设计，可以通过固化手段（C编译、连接、下载和固化）嵌入到产品中。

### ● 可裁剪 (Scalable)

可按需通过裁剪减少产品中μC/OS-II所需的存储器空间（RAM和ROM），可剪裁性靠条件编译实现。

## 亿欧智库： μC/OS-II内核对多任务的控制



# VxWorks: 专业商用级嵌入式操作系统在40年间的升级迭代

VxWorks 是美国 Wind River System 公司（简称WRS，风河公司）于1983年推出的嵌入式实时操作系统，凭借其运行在目标机上的稳定性能、卓越的实时性和商业公司背后专业的支持服务而广泛应用于通信、军事、航空航天等对高精尖技术及对实时性要求极高的领域。在近40年的产品迭代升级中，VxWorks产品已经走过了互联网、多核设备等普及的过程，并正在逐渐向物联网技术转变。

## 亿欧智库：VxWorks作为商业操作系统成功的关键

高性能  
微内核Wind

支持所有的实时特征：快速任务切换、中断支持、抢占式和时间片轮转调度。微内核设计减少了系统开销，从而保证了对外部事件的快速、确定的反应。

可裁剪  
的运行软件

作为商业软件，VxWorks提供了百余种模块以实现不同的配置。允许开发者进行更为优化的 VxWorks 运行环境配置。

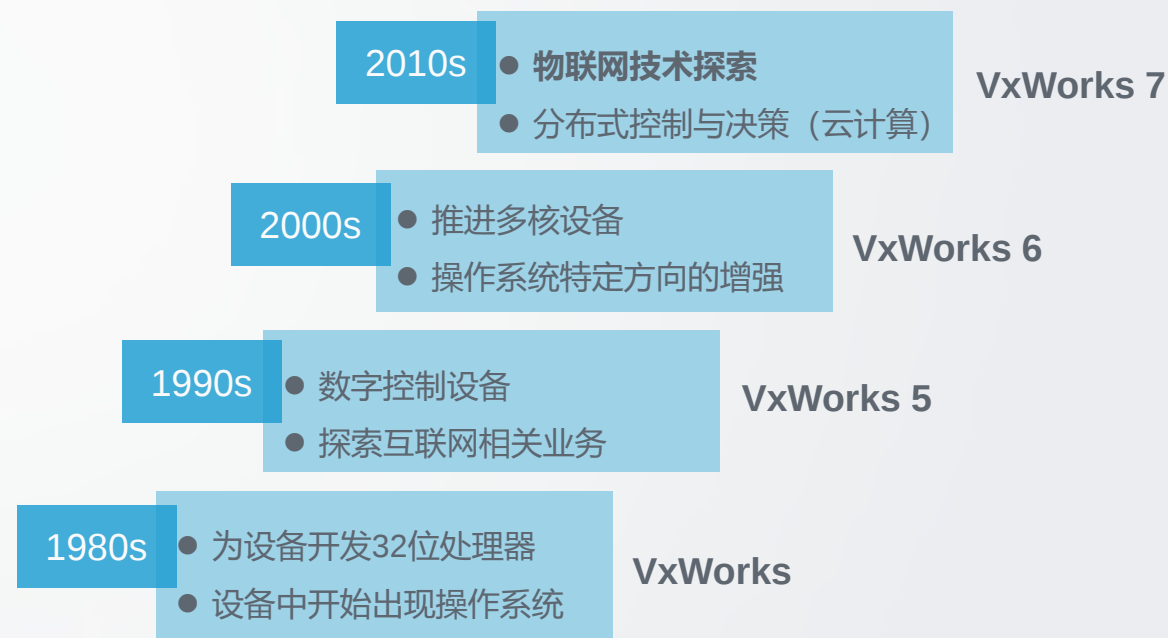
硬件移植  
的优越性突出

VxWorks提供2000 余个现成的板极支持包BSP；并提供 BSP Developer's Kit，用于改变在移植时依赖于硬件的底层代码。

网络支持服务  
的领先

VxWorks 是第一个支持工业标准 TCP/IP 的实时操作系统，并在对网络协议的更新支持上保持领先。

## 亿欧智库：VxWorks四十余年的产品迭代



# VxWorks 7平台：面向物联网时代的全面升级

**VxWorks 7平台的推出，预告着风河公司正式向物联网产业进行新布局。**风河公司产品管理副总裁Dinyar Dastoor曾表示：“物联网显著改变了嵌入式行业格局。今后的RTOS不仅必须具备传统的核心实时、确定性性能和可靠性，还必须提供高度互联、全面安全、远程管理等新能力”。除了在新平台中加入物联网功能外，VxWorks 7核心平台还在探索物联网细分场景的需求。

## 亿欧智库：VxWorks 7平台面向物联网增加的新功能

### 模块化

用户能够对系统组件和协议实施高效且有针对性的升级，无需改变系统内核，从而最大限度地减少了测试和重新认证的工作量。

### 连接性

支持各种业界领先的标准和协议，如USB、CAN、Bluetooth、FireWire和Continual以及开箱即用的高性能组网功能。

### 跨终端性

VxWorks 7平台将微内核与标准内核融为一体，使用户能够在不同类别的设备上运用同一个RTOS基础，适用范围从小型可穿戴设备到大型组网设备。

### 安全性

全套内置安全功能，包括安全数据存储、防篡改设计、安全升级、可信任引导、用户以及策略管理。

## 亿欧智库：VxWorks 7核心平台面向物联网场景的拓展

### 消费领域

启动快、内存占用小，适用于存储空间有限的设备，如移动手持设备、电子成像、住宅网关以及宽带接入等。

### 工业领域

为工业设备制造商提供必要的多媒体联通性中间设备，包括车间驱动程序、连接设备协议、无线外围设备等。

### 医疗领域

平台提供的技术方案可满足医疗设备制造商的特殊要求，最高可供三类医疗设备使用，通过即用安全性与稳定性功能来减少对开发者的挑战。

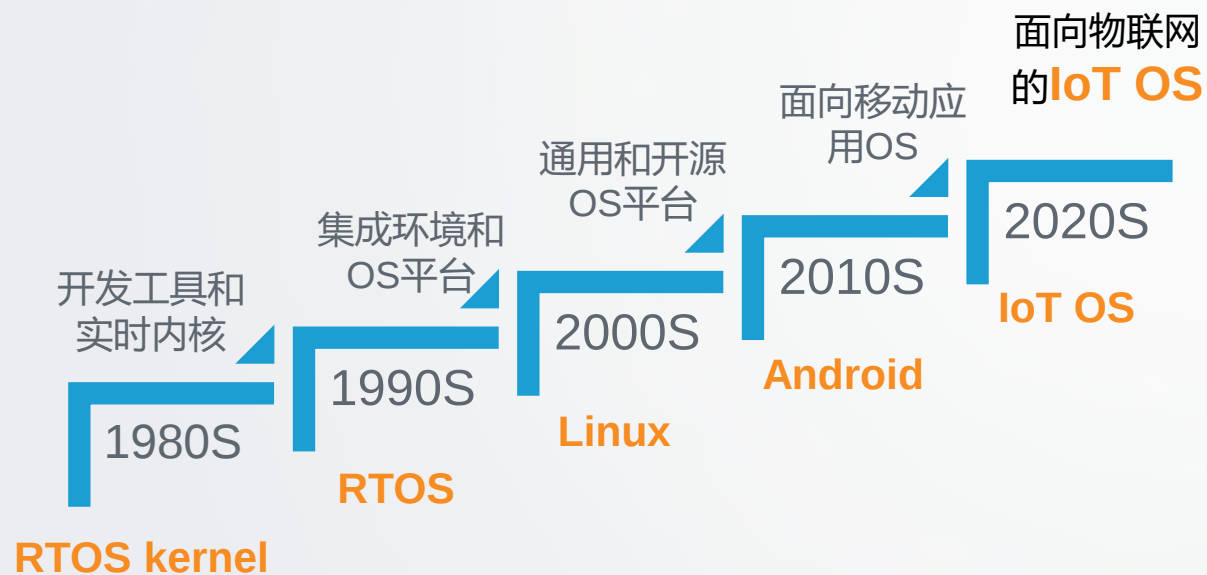
### 网络领域

让客户快速创建、检测、应用、维护和管理高品质网络（有线和无线）基础设施，并提供安全协议来保护网络数据，适用于无线基础设施、企业网络、核心网络、网络边界、LTE和宽带接入设备。

# 小结：嵌入式操作系统每10年经历一次重大技术革命，下一个10年的发展方向是为物联网提供软硬件连接支持

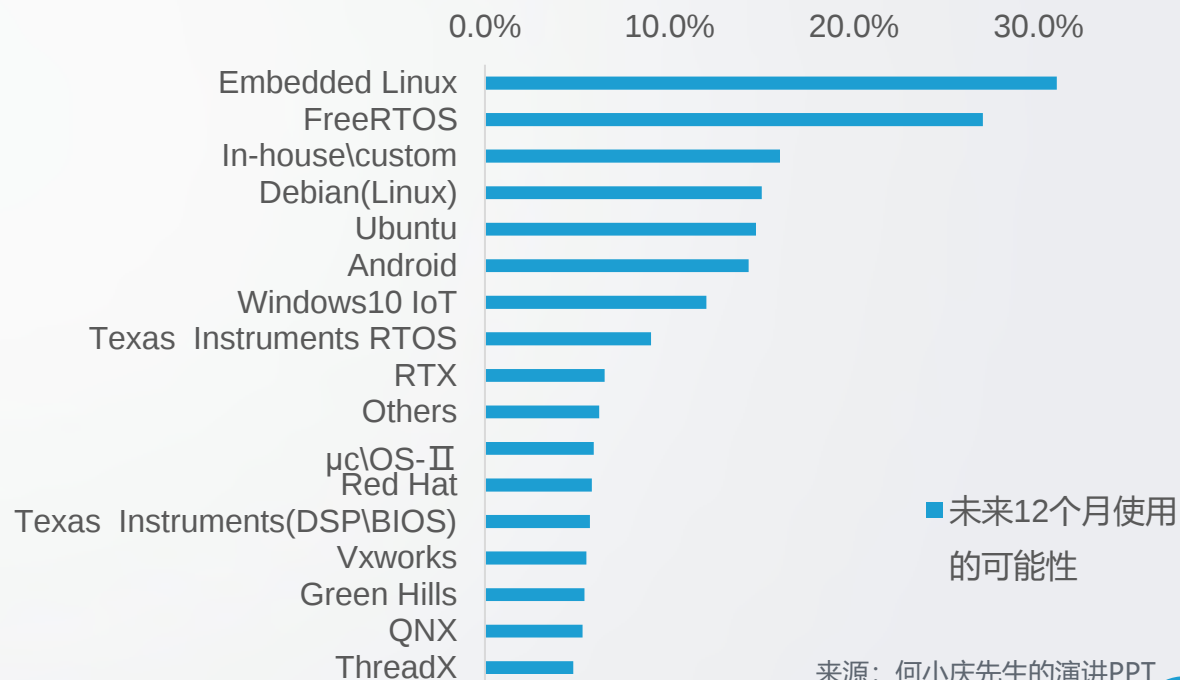
每隔10年嵌入式操作系统都会经历一次重大的技术革命。80年代开始出现实时操作系统的内核，到了90年代这些操作系统逐渐开始丰富自身，出现更多的组间和集成开发环境。2000年之后，通用操作系统开始进入嵌入式领域，Linux在嵌入式领域的影响力显现。2010年左右，基于Linux内核的Android（包括μclinux）开始在消费类产品中占据最重要地位。2020年之后，面向物联网的操作系统（IoT OS）会逐渐脱颖而出，在应用场景的丰富性和市场占有率上有大幅提升。

亿欧智库：嵌入式系统每隔10年的演进



来源：何小庆先生的演讲PPT

亿欧智库：2019年嵌入式OS使用市场调研情况



来源：何小庆先生的演讲PPT

---

## 通用OS裁剪后的操作系统

Tailored operating system

---

# 基于PC OS裁剪后的IoT OS (1/2) : Windows10 IoT Core, 可兼容Windows应用及其他开发和管理工具

基于Linux、iOS、Windows等操作系统的裁剪，代表系统有win10 IOT、Android Things、 WatchOS等。

Windows 10 IoT是较低成本（计算及存储限制）物联网应用下的Win10优化版本，主要包括Win10 IoT核心版和Win10 IoT企业版。

**Windows 10 IoT 核心版**针对带显示屏或不带显示屏的小型设备进行了优化，可以被用来读取传感器数据、控制传动装置、连接到云、创建 IoT 应用程序等。 **Win10 IoT企业版**是完整版本的 Windows 10，可以为企业IoT 解决方案提供可管理性和安全性，兼容Windows 应用及其他开发和管理工具，和PC设备上极为相似。

## 亿欧智库：Windows 10 IoT的三种版本

### Windows 10 IoT 企业版

应用于零售、制造、医疗、金融等行业设备上

### Windows 10 IoT 移动企业版

为需要企业级安全和可管理性的业务线移动设备提供 Windows 10 移动企业版的功能

### Windows 10 IoT 核心版

针对较小的低成本行业设备进行了优化的 Windows 10，专为支持 IoT 网关或微型自助服务终端等设备，同时支持 ARM 和X86架构

Windows 10 IoT for industry devices

最小资源需求：  
1GB RAM、16GB ROM;  
支持x86\x64

Windows 10 IoT for mobile devices

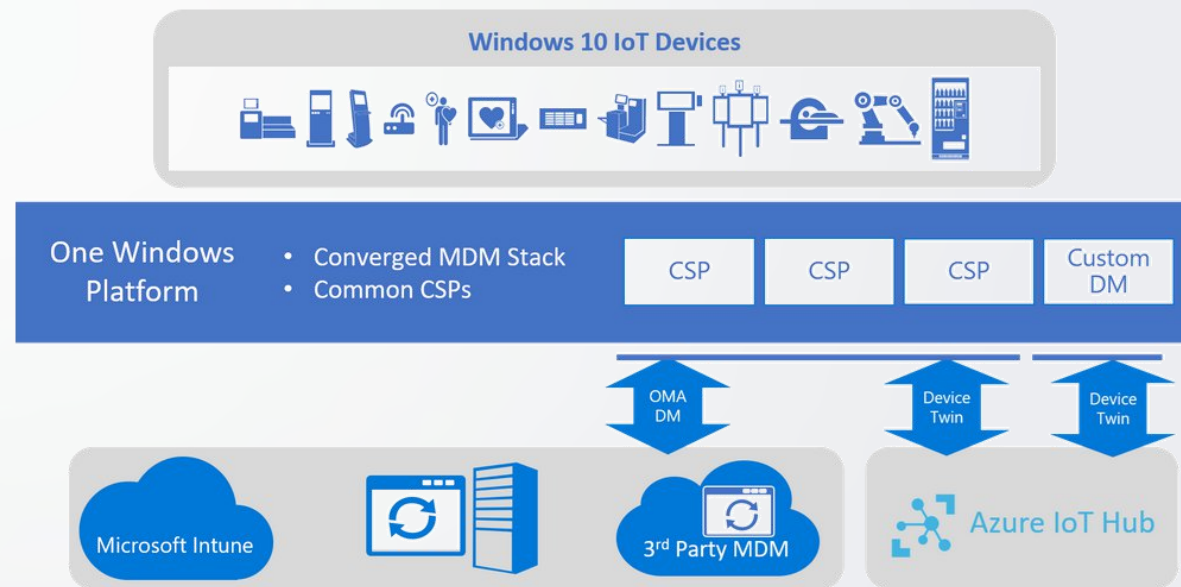
最小资源需求：  
512 MB RAM、4GB ROM;  
支持ARM

Windows 10 IoT Core

最小资源需求：  
256 MB RAM、2GB ROM;  
支持X86\X64 和 ARM

来源：Win10 IoT 官网

## 亿欧智库：Win10 IoT和Azure IoT云之间的联系



基于Azure为Win10 IoT提供云资源服务

来源：Win10 IoT 官网

# 基于移动操作系统裁剪后的IoT OS (2/2) : Android Things, 可利用现有生态为物联网提供新的APIs

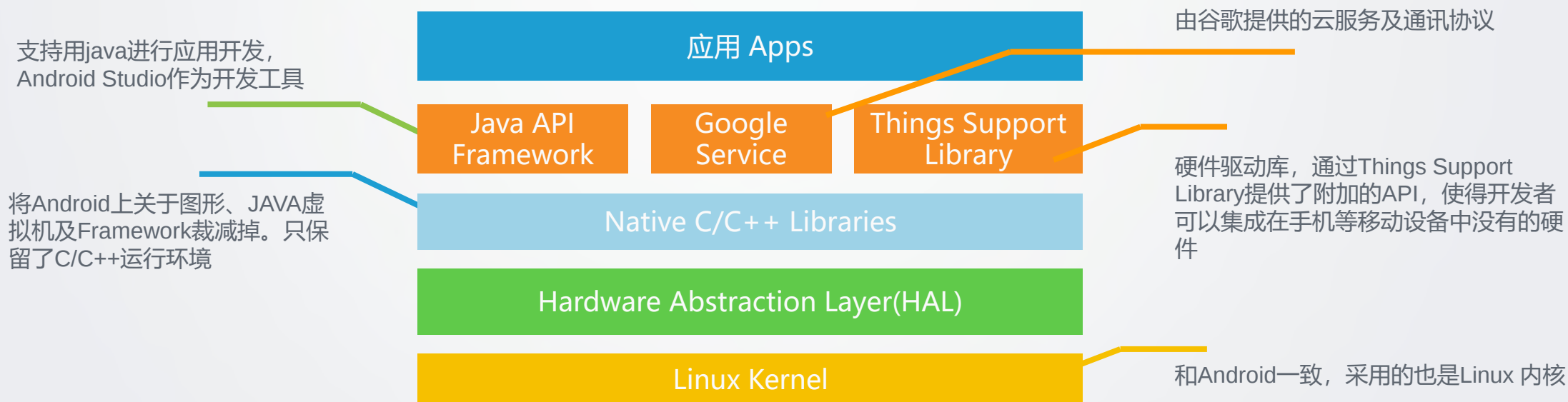
2014至2019年间谷歌在物联网应用布局上进行了很多尝试，其中很重要的两个工具是由Android裁剪而来的Android Things和Android Wear（后改名Wear OS）。Android Things**应用于各种物联网设备**，而Wear OS则**应用于可穿戴设备**。

**Android Things**是 Google 针对物联网和智能硬件推出的一款操作系统，谷歌的理念是通过 Android Things 让 Android 开发者与公司能够以他们过去的开发方式继续开发物联网硬件设备，避免不必要的麻烦。

Android Things 可以依靠现有的生态资源，如开发工具、Android SDK 和庞大的开发者社区等；

Android Things 将为物联网设备提供新的APIs，如低电平I/O接口和为通用组件提供了库支持，如温度显示器、显示控制器等；利用 Android OS 定期的更新机制保证其自身的安全性。

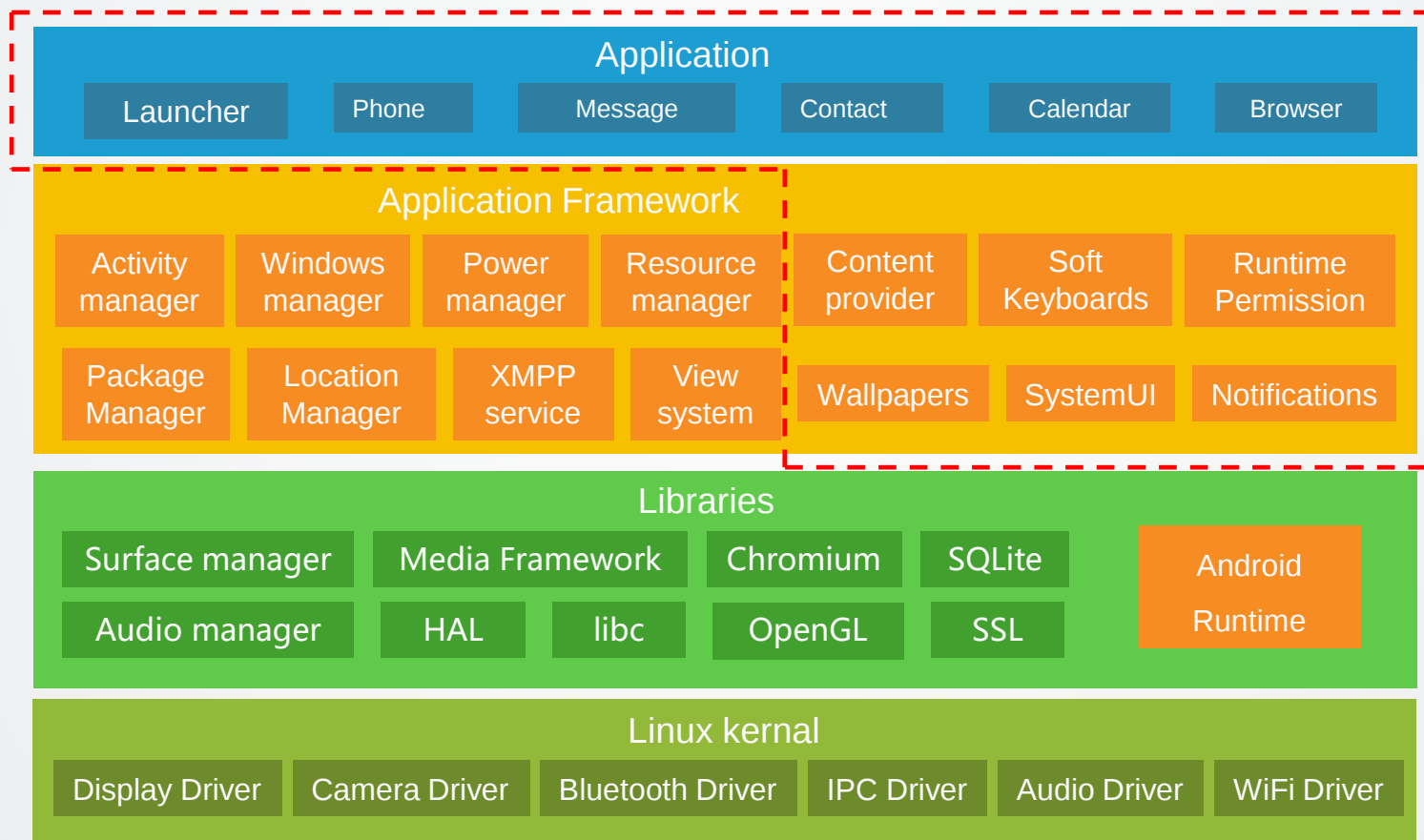
亿欧智库：Android Things系统框架



# 基于移动操作系统裁剪后的IoT OS (2/2) : Android Things , 相较于Android系统, 弱化了交互

将裁剪后的Android Things和Android进行对比可以发现, Application和Application FrameWork层内和交互相关的模块被裁剪掉, Libraries和内核层则没有任何裁剪。这说明Android Things在用户的交互需求上较弱, 主要偏向于处理物联网数据, 因此对上层多应用并没有过多支持。

亿欧智库: Android Things相比于Android 裁剪掉的部分



红框内是Android Things相比于Android 裁剪掉的部分

## 小结：通用系统裁剪后的IoT OS的优势在于软硬件生态成熟，开发难度较小；劣势在于应用场景受限，需要其他系统进行辅助

由成熟系统裁剪而来的IoT OS优势在于，其开发模式可以接近于成熟系统，比如Android Things以Java语言开发，使用Android Studio作为开发工具，开发过程和Android几乎相同。但**裁剪后的IoT OS显然在应用场景上有限**，例如微软在推出基于Win10 的IoT版本后，又相继推出Azure Sphere和Azure RTOS作为补充。

亿欧智库：轻量级IoT OS和裁剪后的OS对比

| 特征   | 轻量级IoT OS                       | 裁剪后的IoT OS                            |
|------|---------------------------------|---------------------------------------|
| 是否开源 | 绝大多数都开源                         | 闭源                                    |
| 主导公司 | 各色背景                            | 传统商业操作系统公司                            |
| 适用硬件 | 主要适配 MCU<br>内核最小占用RAM可<br><10Kb | 主要适配 MPU<br>对硬件要求较高，内核占用RAM<br>可达几十Mb |
| 上云支持 | 重视，厂商基本都提供第<br>三方上云支持           | 有偏向性，主要和自有云平台对<br>接（如Azure和谷歌云）       |
| 软件开发 | 需要培养开发者社区                       | 基本和原有开发者习惯一致，较<br>完美迁移                |
| 核心优势 | 实时性更好；更符合低功<br>耗需求              | 具有开发者基础；目前可在端侧<br>较顺利运行AI算法           |

### 核心结论：

- 传统商业操作系统公司从裁剪现有系统出发，以迎合物联网场景下的需求，在内核技术上更少体现硬实时性；
- 裁剪后的IoT OS优势在于开发语言和工具上沿袭了传统，使得开发者可以按照原有习惯进行软件开发；
- 裁剪后的IoT OS往往都会和自有的AI技术相结合，在端侧输出AIoT能力；
- 裁剪后的IoT OS仅能满足物联网场景下的小部分终端设备需求，往往需要和其他操作系统一起配合。

---

# 轻量级物联网操作系统

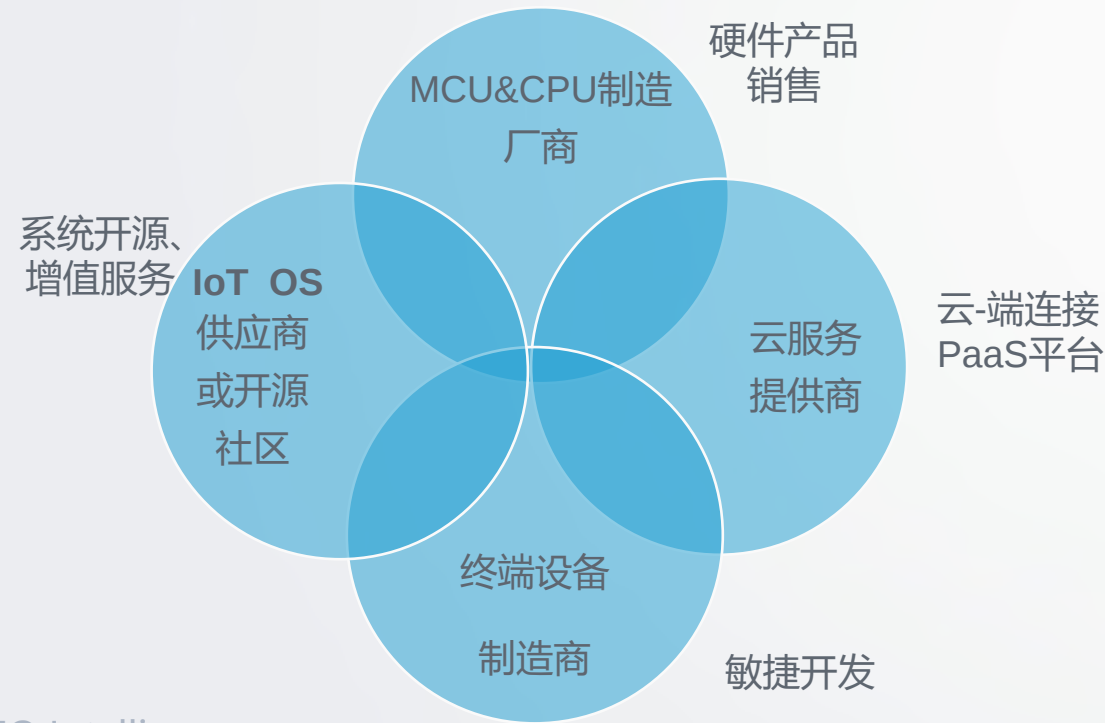
Lightweight IoT operating system

---

# 轻量级物联网操作系统——满足终端物与物实时连接

物联网开发场景下大量使用嵌入式系统，并添加物联网场景所需要的的通讯协议栈、中间件和组件。轻量级物联网操作系统很好的满足了低功耗和低计算能力下对MCU的限制。轻量级IoT OS基本都为实时内核，其中的代表操作系统有FreeRTOS、RT-thread、mbed OS、LiteOS等。此类操作系统的优点在于，和底层硬件的适配较好，缺点则在于上层软件开发遇到种种问题，应用拓展困难。云服务商提供的RTOS可以让用户更快地开发，加速物联网项目的落地。

亿欧智库：IoT OS和硬件厂商的关系



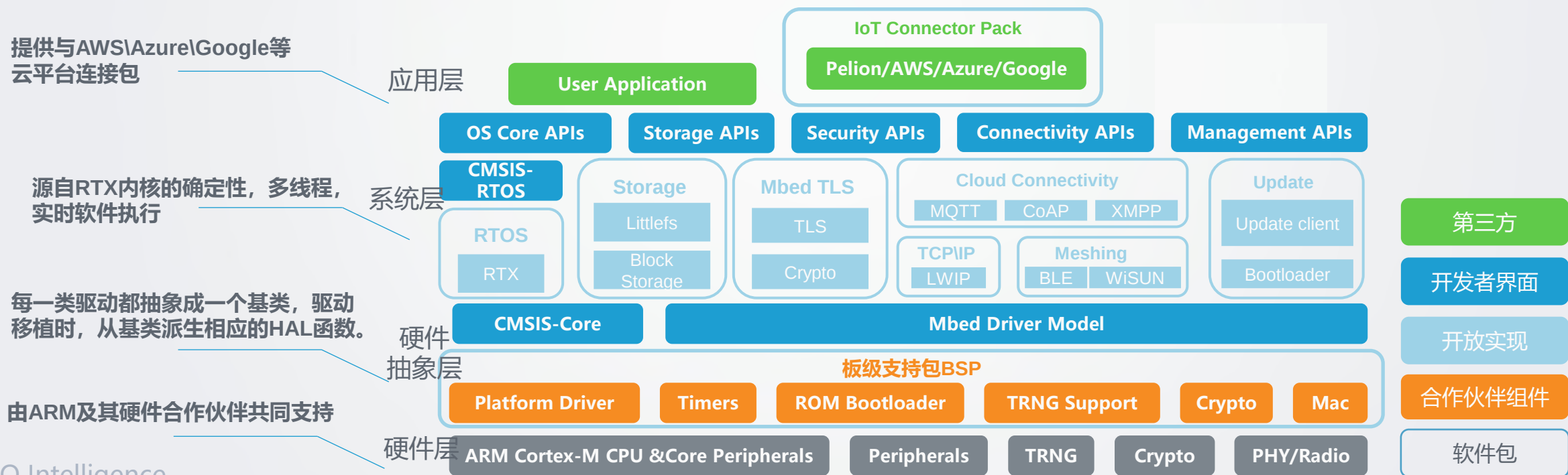
亿欧智库：影响轻量级IoT OS性能的6大关键因素

- **内存占用:**  
由于IoT终端设备资源有限，因此OS占用的内存有限；
- **可扩展性:**  
OS的基本作用是隐藏不同硬件的细节差异，因此需要考虑对于不同硬件的支持性；同时，除强制性内核外，其他功能应该以附件组件的形式存在；
- **可移植性:**  
OS将底层硬件和应用程序隔离开来，需要以标准形式（例如兼容POSIX）移植到不同的硬件平台；
- **连接性:**  
OS需要支持不同种类的连接协议，以适应连接方式的多样性；
- **安全性:**  
OS需要以安全启动，SSL支持，组件和加密驱动程序保证安全；
- **可靠性:**  
OS自身及对其应用程序应该有一套认证模式，以确保长期远程工作中不出现故障。

# Arm Mbed OS: 硬件广泛兼容, 软件易于移植开发

**Arm Mbed OS**是由Arm公司推出的开源IoT操作系统。Arm 的生态系统已经产出数十亿个由 Arm合作伙伴出货、基于 Arm 架构的芯片所带动的 IoT 装置。为进一步高效地推进IoT硬件市场, Arm积极推动操作系统及 Mbed 操作系统伙伴管理(Mbed OS Partner Governance)模型。Mbed OS的优势在于, 从硬件层架构设计入手, 在硬件兼容性规范上具有先天优势。ARM在Mbed平台上提供了60多经过Mbed Enable认证的硬件方案供开发者选择, 包括100多种开发板和400多个元件库, 涵盖了市面上主流的芯片厂商, 如NXP、Atmel、ST、瑞萨等, 减少了软件开发者因为硬件不兼容原因的二次开发和移植困境。

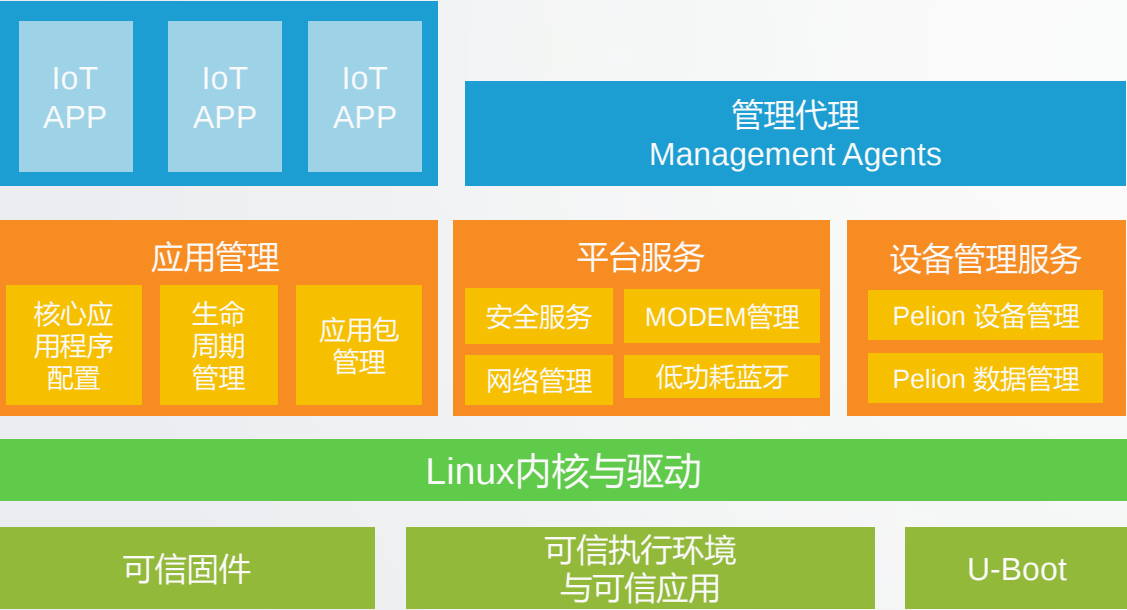
亿欧智库: Arm Mbed OS 结构框架



# Arm Mbed Linux OS：在Mbed OS基础上整合 Linux Kernel 及 Yocto 方案，增加服务可用性

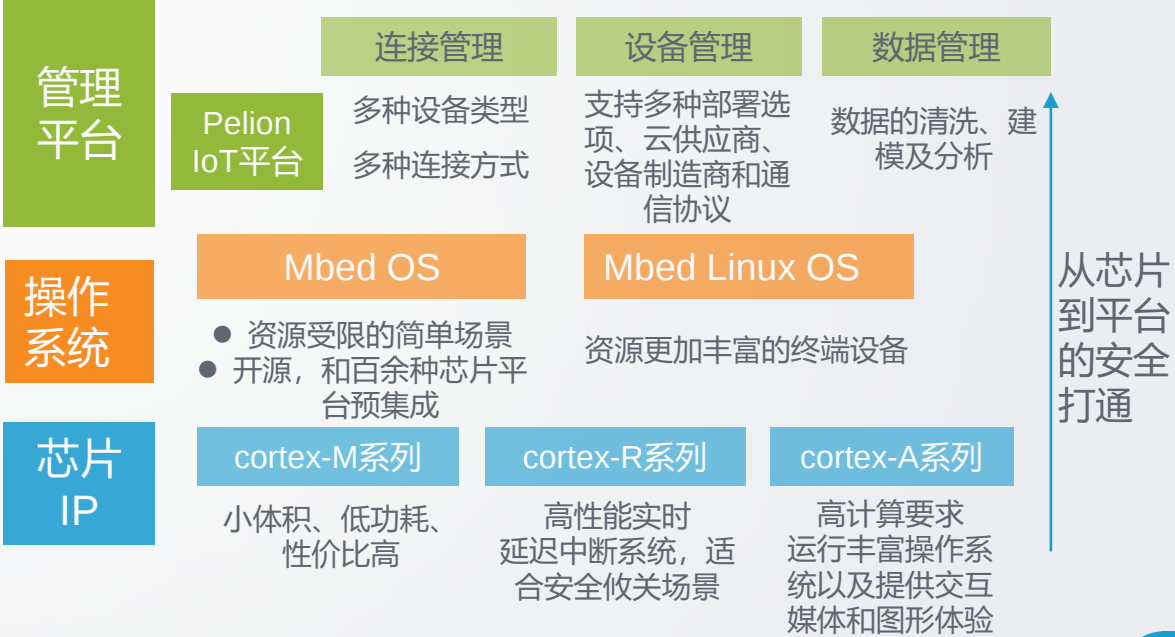
**Mbed Linux OS** 是Arm公司近年来推出的第二款物联网操作系统。如果说Mbed OS面向的是资源受限，希望用简单快捷的方式满足连接、安全以及应用开发需求的设备，那Mbed Linux OS则面向网关、摄像头、工业电脑这类资源配置更丰富的设备，方便用户集中开发高附加值的功能。ARM是一家专注于底层技术的公司，认为操作系统和管理平台作为物联网时代的基础。从其物联网布局可以看出，广义的物联网操作系统是ARM最为关注的部分，无论是两套适应不同应用场景的操作系统，还是基于多终端管理需求出发的IoT平台，ARM希望从底层技术出发掌握物联网产业的发展重心。并且，打通芯片到云端也更利于物联网安全。

亿欧智库： Mbed Linux OS 结构框架



来源：Mbed Linux OS 官网

亿欧智库： Arm以操作系统、管理平台为核心的底层技术构建

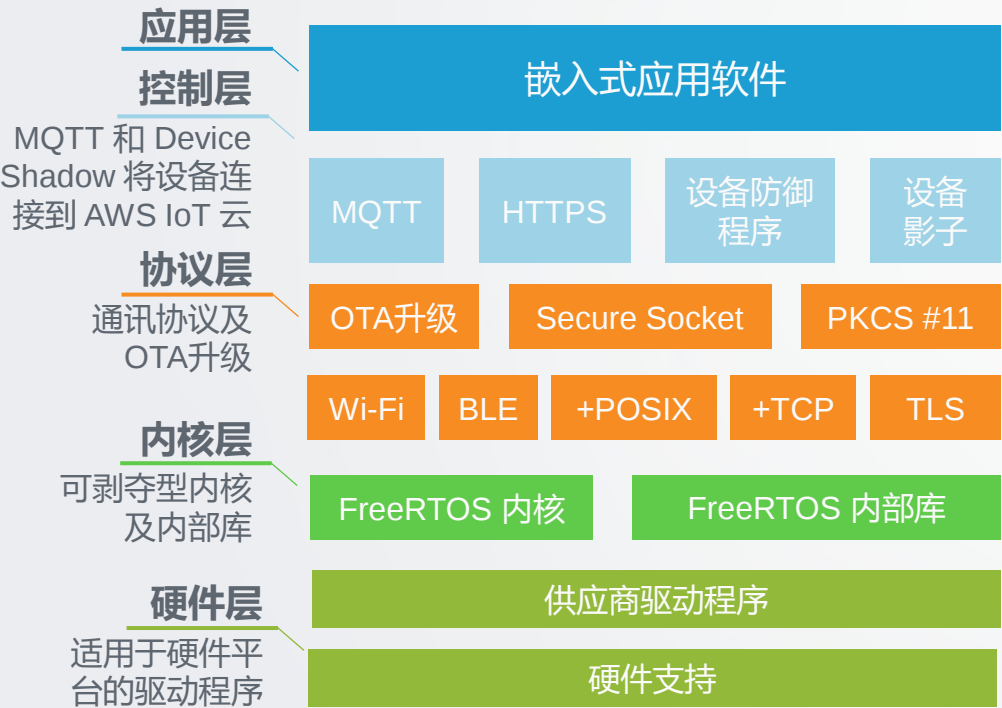


来源：亿欧智库整理

# Amazon FreeRTOS：内核灵活无限制，实时响应低时延，可广泛支持小型设备

**FreeRTOS** 作为2003年推出的老牌开源嵌入式操作系统，遵循MIT开源许可发布，在全球范围内拥有大量使用者。据EEtimes杂志嵌入式系统市场报告2011-2017数据显示，FreeRTOS在RTOS内核使用榜和RTOS内核计划使用榜上都居前列。2017年之后FreeRTOS还获得了亚马逊平台的支持，使得低功耗的小型设备可以安全地连接到 AWS 云服务上，提高端侧操作系统的云端支持能力。获平台加持后的Amazon FreeRTOS 构建侧重于设备可靠性和易用性，具有可裁剪、可剥夺型内核，且没有任务数量限制。

亿欧智库：Amazon FreeRTOS框架及生态建设情况



| AWS IoT Core | AWS IoT Device Management | AWS IoT Device Defender | AWS IoT Analytics |
|--------------|---------------------------|-------------------------|-------------------|
| 云应用程序与设备交互   | 注册、组织、监控、远程管理 IoT 设备      | 保护 IoT 设备队列安全           | 非结构化数据分析          |

|                                  |                          |
|----------------------------------|--------------------------|
| <b>AWS边缘计算平台 Greengrass Core</b> | 实时响应本地事件，通过容器支持实现简化的设备编程 |
|----------------------------------|--------------------------|

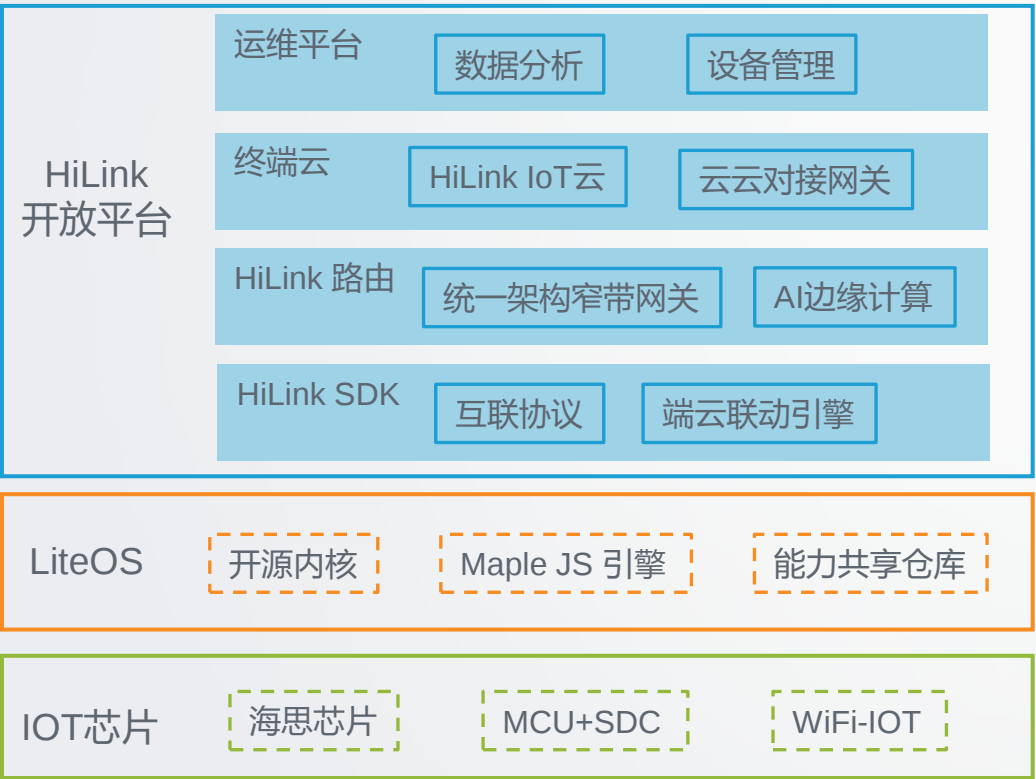


- 应用场景 (Application Scenarios)**
- 消费品 (Consumer products)
  - 能源 (Energy)
  - 医疗保健 (Healthcare)
  - 工业 (Industry)
  - 零售 (Retail)
  - 安全 (Security)
  - 智能城市 (Smart cities)
  - 智能家居 (Smart homes)
  - 交通 (Transportation)

# Huawei LiteOS：上承HiLink开放平台，下接华为自研芯片，是华为物联网体系的重要一环

**Huawei LiteOS**是华为面向物联网领域开发的**实时内核轻量级操作系统**，于2015年正式推出，并于2016年开源。LiteOS可广泛应用于智能家居、个人穿戴、车联网、城市公共服务、制造业等领域。LiteOS是华为物联网战略中的重要一环，上承**HiLink开放平台**，下接华为自研及第三方提供的**物联网芯片**。

**亿欧智库：LiteOS是华为物联网体系解决IoT生态建设难题的重要一环**



# Huawei LiteOS：异构芯片、轻量内核、OpenCPU决定其易开发、低功耗、端云互通组件完备的优势

LiteOS采用了OpenCPU架构，MCU和通信模组二合一，降低终端设备大小和成本；基于物联网领域业务特征打造领域性技术栈，为开发者提供“一站式”完整软件平台；SDK端云互通组件预集成了全套IoT互联互通协议栈，有效降低开发门槛、缩短开发周期，能够帮助众多行业客户快速的推出物联网终端和服务。

LiteOS具备轻量级、安全性、低功耗、互联互通、组件丰富、快速开发等关键能力，具有轻量级、联接智能、组网智能和管理智能等主要特点。

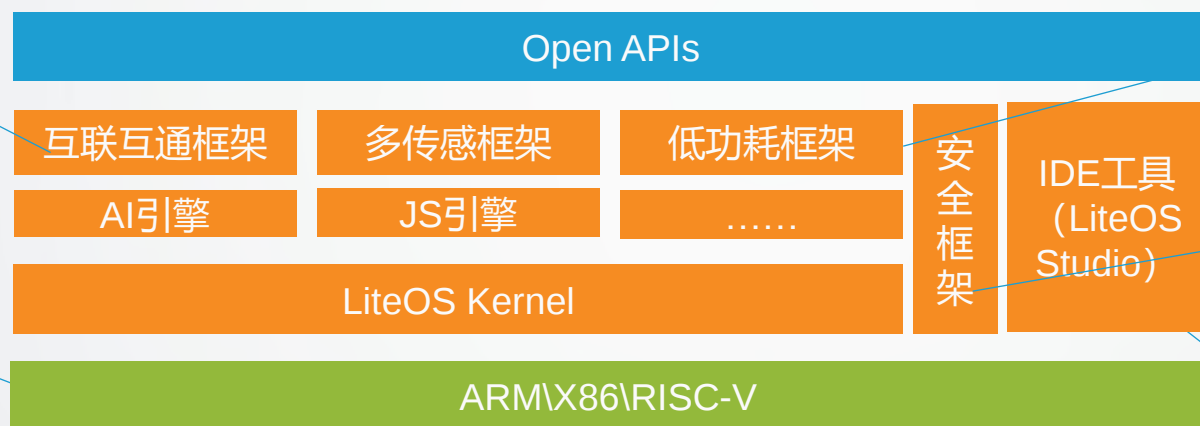
亿欧智库：LiteOS的产品架构及优势

## 互联互通

集成了 LwM2M、CoAP、MQTT、mbed TLS、LwIP等全套IoT互联互通协议栈，大大减少开发周期

## 异构芯片

支持ARM Cortex-M等系列芯片架构，适配了30+ 通用 MCU开发套件



## 低功耗

最小内核尺寸仅为6KB，具备快速启动、低功耗等优势，Tickless机制显著降低传感器数据采集功耗

## 安全性

构建低功耗安全传输机制，支持双向认证、FOTA固件差分升级，DTLS/DTLS+等，构建低功耗安全传输机制

## IDE工具

集成开发环境，一站式开发工具，支持C、C++、汇编等语言

### 【轻量级】

- 可伸缩内核
- uA级功耗
- uS级响应

### 【联网智能】

- WiFi、Zigbee、NB-IoT等协议的互通

### 【组网智能】

- 优化的Mesh自组网，组网稳定且快

### 【管理智能】

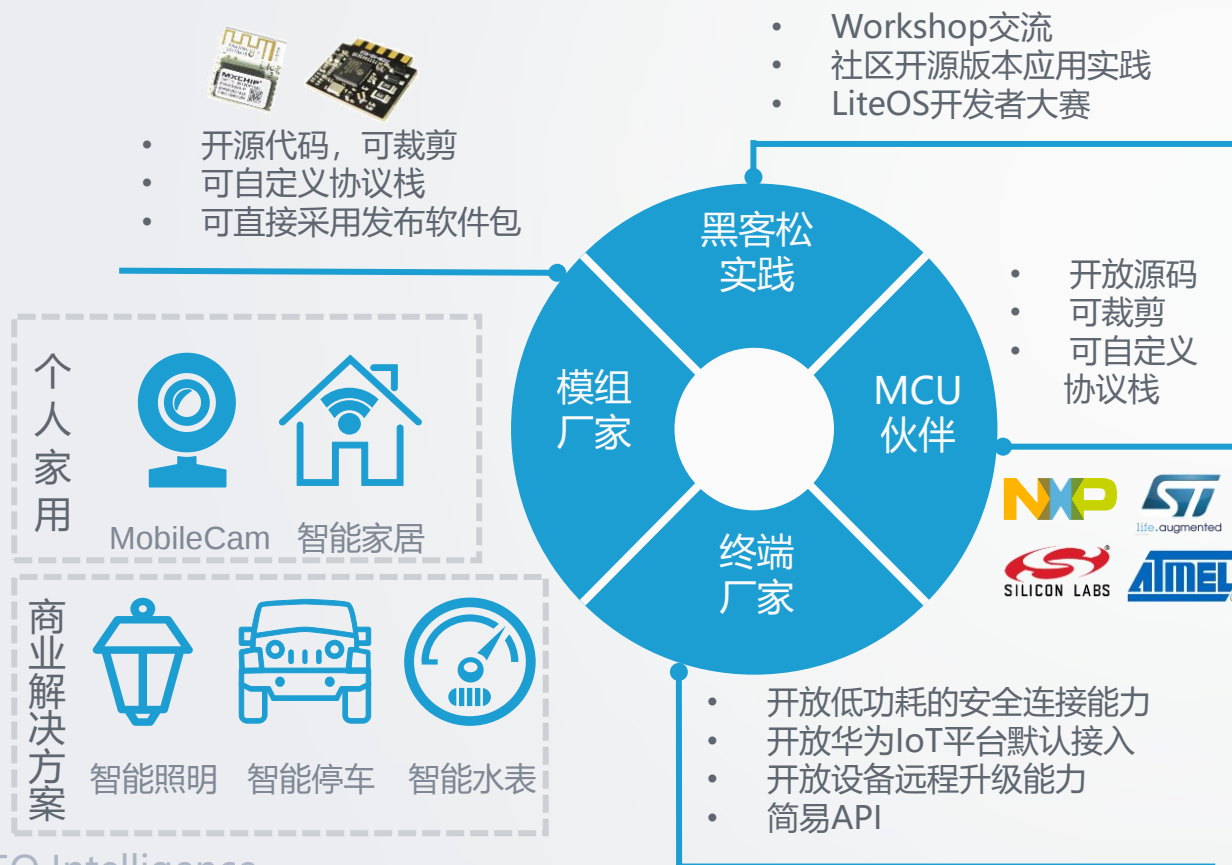
- 不同类型、不同接口传感器的统一管理，即插即用

来源：华为LiteOS官网

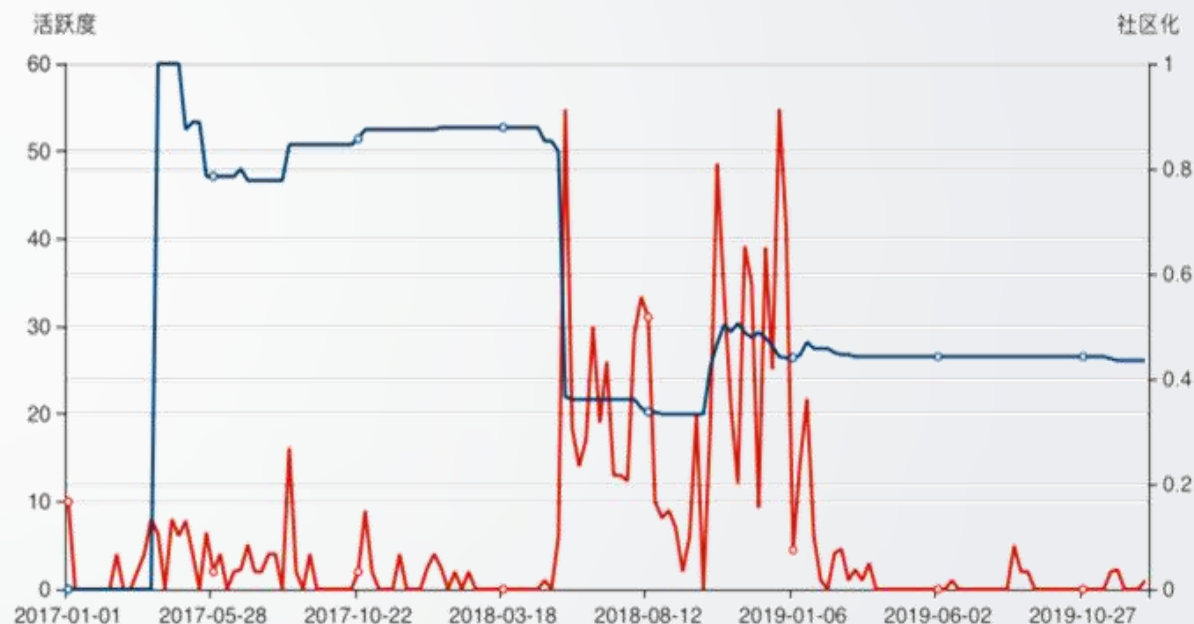
# Huawei LiteOS: 开发者生态的构建

LiteOS于2015年发布，遵循 BSD-3 开源许可协议，2018年是其发展高峰期，2019年以后的发展趋势趋于低缓。从Grank的结果可知，LiteOS的用户类型较为集中，主要应用在智能家居、智能穿戴、车联网、城市公共服务等领域。

## 亿欧智库：LiteOS的开放能力及应用情况



## 亿欧智库：LiteOS在开源社区的活跃度数据及社区化程度



数据来源：Grank (GitHub Rank)

注：活跃度数据为项目的提交数、拉取请求数和贡献者数加权数  
社区化程度以实际向项目贡献代码的人员社区化离散程度来评估

—— 社区化程度  
—— 社区活跃度

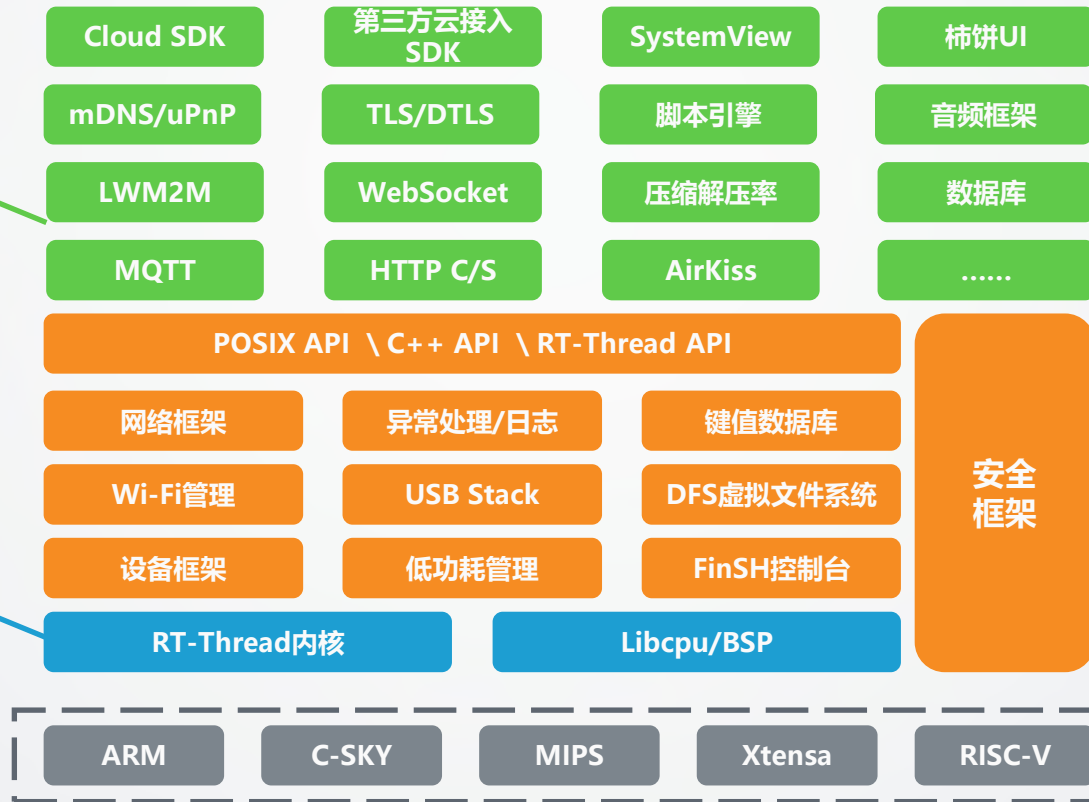
# RT-Thread：来自开源社区，可根据编程对象实现模块化裁剪

RT-Thread2006年诞生于中文开源社区中，以满足当时对小型实时操作系统（RTOS）的更高需求。经过10余年的发展，RT-Thread已经成长为组件丰富、高度可伸缩、开发简易、低功耗、安全保障的物联网操作系统，被广泛应用于能源、汽车、消费电子等多个行业，截止2020年累积装机超过6亿台。RT-Thread的广泛应用离不开其基于Apache License 2.0 开源许可协议的免费商用和丰富组件支持。

亿欧智库：RT-Thread系统架构图

**软件包。**运行于 RT-Thread 物联网操作系统平台上，面向不同应用领域的通用软件组件，由描述信息、源代码或库文件组成。具有强可重用性，模块化程度高。

**内核层。**包括内核系统中对象的实现，例如多线程及其调度、信号量、消息队列、内存管理、定时器等；libcpu/BSP（芯片移植相关文件 / 板级支持包）与硬件密切相关，由外设驱动和 CPU 移植构成



**组件和服务层。**基于 RT-Thread 内核之上的上层软件，例如虚拟文件系统、FinSH 命令行界面、网络框架、设备框架等。采用模块化设计，做到组件内部高内聚，组件之间低耦合。

**对主流芯片支持较好。**主要支持 MCU 芯片，包括 ARM Cortex\MIPS\RISC-V等。部分系列的CPU在特定场合下也可使用。

# RT-Thread：物联网多终端场景的布局

RT-Thread以提供专业开发板、RT-Thread Studio开发工具、图形化UI控件、多样性的传感器支持包形式来扩大自身操作系统的行业影响力。从其应用领域来看，带屏设备及无屏设备皆有覆盖，创新型智能家电是其重点应用的领域。

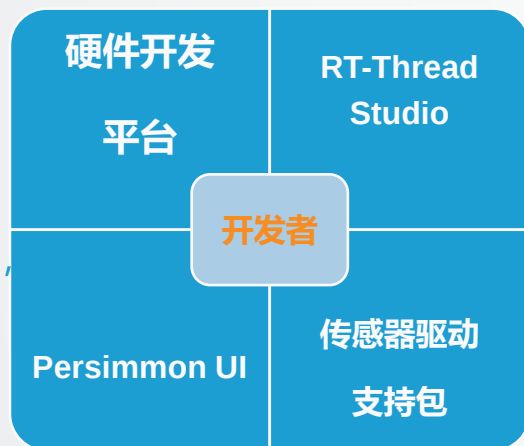
## 亿欧智库：RT-Thread系统提供给开发者的软硬件工具

- MCU类开发板
- IoT开发板
- 音视频类开发板



图形用户组件界面，用于提供图形界面的用户交互

- 所见即所得
- 提高UI开发灵活性
- 降低开发成本



包括工程创建和管理，代码编辑，SDK管理，RT-Thread配置，构建配置，调试配置，程序下载和调试等功能

应对物联网应用极度碎片化，解决物联网终端厂商、系统集成商和方案商在面对传感器时的困境

来源：亿欧智库整理

## 亿欧智库：RT-Thread的部分应用案例



智能故事机

教育娱乐机器人，可实现无按键唤醒词交互



共享充电宝

使用蓝牙+2G双重通讯，不受场景WIFI不稳定影响



脑电波意念机

通过采集、分析大脑的实时脑电波，感知到事物对人产生的状态



充电桩

壁挂立柱兼容式交流充电桩，带4G联网功能



医用空气净化器

基于空气低温等离子体消毒技术达到医用级杀菌效果



智能无线吸尘器

可以在主机端或手机端手动调节功率，或由检测到的吸入灰尘浓度自动调节功率



微信收款音箱

基于腾讯云物联网开发平台，统一管理多设备的接入



手环

基础功能全面，支持 Alexa 语音助手

来源：亿欧智库整理

# 亿欧 几种轻量级物联网操作系统的详细参数比较

在具体的参数和指标对比上，何小庆先生曾经在论文中针对Huawei LiteOS的2018年3月30日 C50社区开源版及LiteOS SDK10.0、Amazon FreeRTOS V1.2，还有RT-Thread V3.1.1做过参数比对，具体结果列于下表。

亿欧智库： 三种IoT OS 的参数

| 特征                  | Huawei LiteOS SDK10.0       | Amazon FreeRTOS V1.2                   | RT-Thread V3.1.1             |
|---------------------|-----------------------------|----------------------------------------|------------------------------|
| License             | BSD                         | MIT                                    | APACHE                       |
| Architecture        | ARM Cortex-M                | Many MCU                               | ARM\X86\RISC-V MCU &MPU      |
| Program language    | C\C++ EXT JS                | C\C++                                  | C\C++ EXT JS , python        |
| IDE                 | KEIL\IAR                    | KEIL\IAR\CCS\GCC\MPLAB                 | KEIL\IAR\GCC                 |
| POWER MANAGEMENT    | Kernel Tickless             | Kernel Tickless                        | RT-Thread PM                 |
| Program Mode        | Multi-Threading             | Multi-Threading                        | Multi-Threading              |
| Wireless Connection | WiFi\NB-IoT\COAP\LWM2M      | WiFi\MQTT                              | WiFi\BLE\MQTT                |
| Security            | DTLS                        | TLS\Security Socket\Permission         | TLS                          |
| OTA                 | FOTA                        | Full Support and OTA abstraction layer | OTA example                  |
| Footprint           | IOTSDK RAM 32KB FLASH 128KB | Kernel RAM 10KB FLASH 12KB             | Nano RAM 1.04KB FLASH 3.25KB |
| Middleware          | Very few                    | TCP\IP , FAT and many thirdparty       | FAT\JAFFS\CANOPN             |



# 小结：不同背景厂商布局物联网操作系统，采用开源模式扩大市场规模，构建软硬件生态

近年来越来越多的科技互联网公司开始关注于物联网操作系统（IoT OS）的布局，除以上所述的五种外，还有如腾讯于2019年推出的Tencent OS Tiny，微软于2020年推出Azure RTOS，以及中移动推出的OneOS等。我们认为ARM Mbed OS、Amazon FreeRTOS和RT-Thread分别具有代表性，因此重点去进行对比。

亿欧智库： 三种具有代表性IoT OS 对比

| 特征     | ARM Mbed OS                           | Amazon FreeRTOS                         | RT-Thread                               |
|--------|---------------------------------------|-----------------------------------------|-----------------------------------------|
| 是否开源   | 是                                     | 是                                       | 是                                       |
| 主导公司   | 芯片IP领先者                               | 云计算领军企业                                 | 科技创业公司                                  |
| 硬件支持性  | 从硬件底层出发，基本覆盖市面上主流芯片厂商                 | 由于发展历史悠久，安装量大，获得全球顶尖MCU和IoT 芯片厂商的支持     | 不断拓展支持的架构类型，目前已覆盖有ARM\X86\RISC-V等架构     |
| 上云支持性  | 既支持上自有pelion平台，也支持上其他云平台，如中移动OneNet   | 仅支持AWS云平台，官方不提供支持其他云平台                  | 重点支持开放第三方云平台，对阿里云、腾讯云、中移动OneNet都有较好支持   |
| 通讯协议支持 | 支持BLE、WiFi、6LoWPAN、NFC、Thread、NB-IoT等 | 物联网通讯协议支持不足，缺少对2G\3G\4G以及NB-IoT和Lora的支持 | 支持Wifi、蓝牙、NB-IoT、2G\3G\4G等              |
| 发展瓶颈   | ARM本身在云平台建设上不具备优势，IoT战略的贯彻值得长期观察      | 由于官方仅支持AWS云平台的接入，与国内本地IoT生态互动少          | 独立IoT OS开发公司，单纯靠操作系统收费困难，能否有持续的资金用于技术研发 |

来源：亿欧智库整理

## 核心结论：

- 1、基本上所有的IoT OS 都采用开源模式，期望形成自己的软硬件生态，这导致了目前阶段使用物联网操作系统的成本很低。
- 2、芯片架构厂商、云计算厂商和OS领域的初创公司都推出了较为成熟的产品，希望从未来的IoT大市场中占据有利位置。
- 3、IoT OS本身的技术复杂度并不如传统的PC和移动端系统，难点在于对碎片化终端需求的满足。目前IoT OS 还处于混战阶段，未来极有可能出现2-3家市场占有率高的优秀IoT OS。

---

# 下一代操作系统

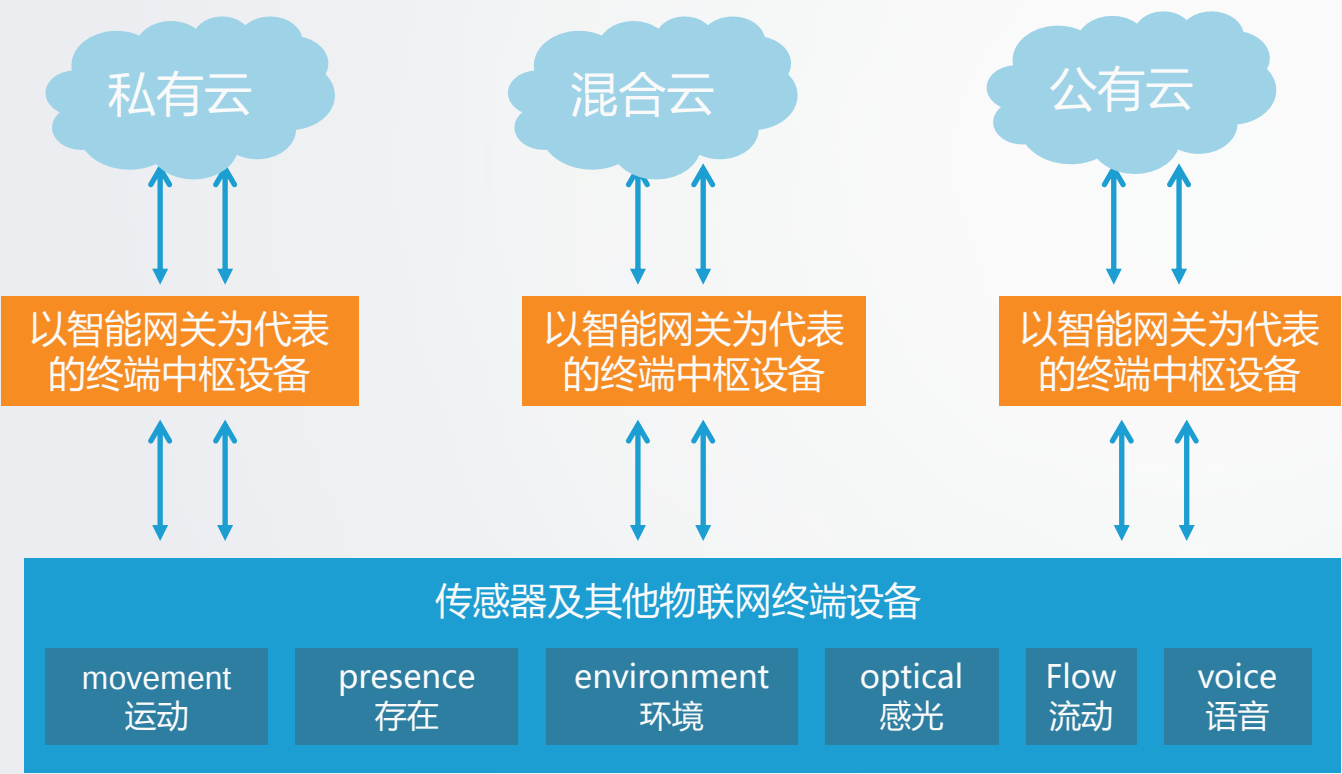
Next generation operating system

---

# 完整的IoT网络需要多种类型的操作系统共同实现，物联网终端碎片化的发展趋势催生了下一代物联网操作系统

物联网硬件的操作系统使用通常分为两类：终端设备和网关。与网关相比，终端设备或节点的功能需求通常要小得多。因此应用于终端中枢设备的系统和其他IoT终端设备有较大的差异，实现的功能目标也完全不同。是否有一种从设计结构上就符合物联网的多终端和碎片化需求的IoT OS可以满足多方面需求？我们关注到谷歌Fuchsia和华为鸿蒙OS可能会实现。下面将重点介绍这两个系统。

## 亿欧智库：一个完整的IoT网络需要多种类型的操作系统共同实现



终端中枢设备往往承担起较重的任务，Linux、Win10 IoT和Android Things大多服务于此类需求：

- 由**中枢设备进行端侧的数据整合、实时分析，并作出实时决策**，减少去往云端的数据量；
- 此外还要起到**保护下游设备不被非法侵入的防火墙功能**；
- **与基于云的物联网平台的可靠通信**，在网络中断的情况下，备份传输以及管理数据和设备；

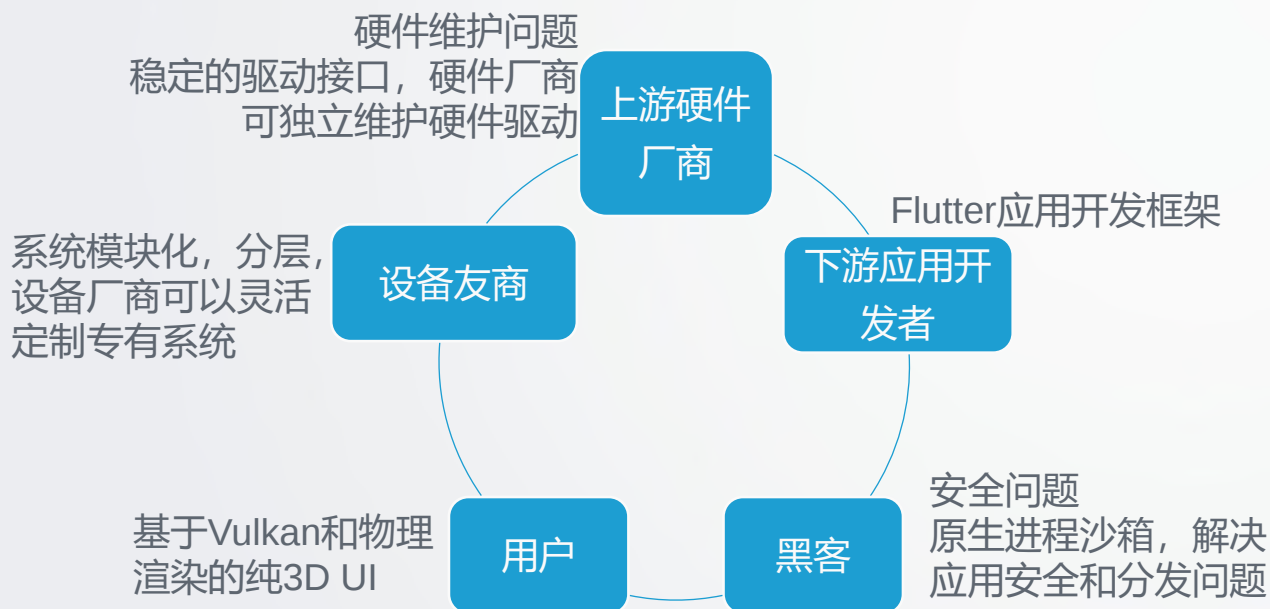
小型IoT终端设备往往采用轻量级的IoT OS，甚至于不采用OS而直接用裸机模式：

- 一般认为，在8位MCU上很少使用OS，消耗少于16KB的RAM和Flash / ROM的系统不需要操作系统；
- 在资源稍高的设备上，采用OS和开发中间件将简化软硬件开发过程，提升设备能力。

# 谷歌：为解决Linux内核与物联网不适配的问题，Fuchsia应运而生，分层模块化的架构更适合裁剪定制，更贴近终端需求

Android及Chrome OS带给谷歌在操作系统上的经验积累，同时也让谷歌认识到基于Linux内核的诸多局限。尽管Android为了弥补Linux的缺陷打了厚重的中间层，但仍无法从根本上解决问题。谷歌Fuchsia项目在设计之初就瞄准了现有OS的诸多缺陷，旨在适用于从手机、PC到智能设备的各种物联网设备。Fuchsia采用Zircon微内核，并使用Flutter作为开发框架，提供了上游硬件厂商、下游应用开发者、设备友商、用户和安全等5方面的改进。在系统架构上，Fuchsia采用模块化和层级化的方式来化解Android的设计缺陷。

亿欧智库：设想中Fuchsia如何解决现代OS痛点



来源：许中兴博士演讲PPT

亿欧智库：Fuchsia系统架构示意图



解耦后的操作系统更适合**裁剪定制**

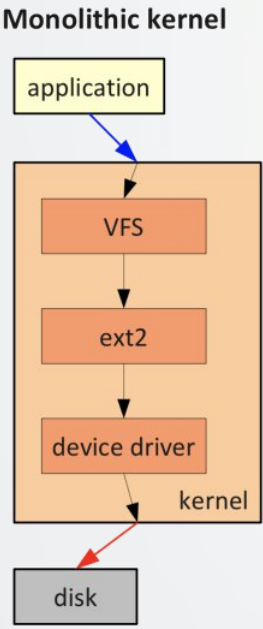
# Fuchsia采用微内核，灵活易用，但是IPC耗能高、效率低的问题亟需改进

由于Fuchsia完全的抛弃了宏内核，其微内核带来的程序运行特点值得一提。微内核将系统分为小的功能模块，只在内核中保留最核心的调度等功能，这也意味着大部分的系统级服务都被拆分到了内核外，以**进程间通信（IPC，Inter-Process Communication）**形式执行。IPC是微内核系统消耗最大的一部分，存在很大的改进空间，是目前阶段的技术难点所在，尤其是对于规模较大的内核而言。

亿欧智库：微内核Zircon相比于Linux宏内核的运行差异

宏内核  
monolithic kernel

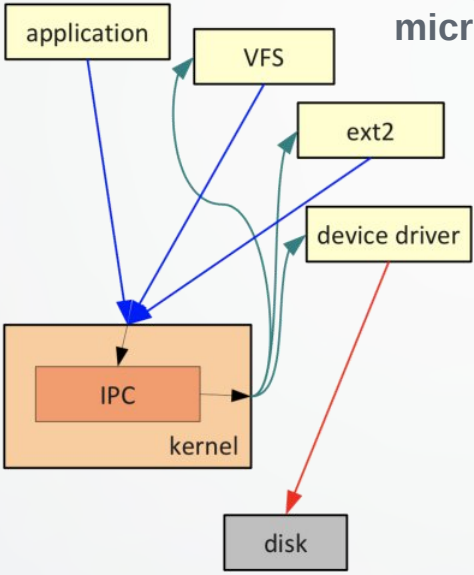
一个任务在内核中被完整的执行，共享内存空间。执行效率更高，速度更快，但安全性无法保证。且因为各功能模块间耦合度高，维护更新难，可拓展性差。



Microkernel

微内核  
microkernel

一个任务被拆分成几个子任务，以进程间通信的形式在内核态外部实现完成



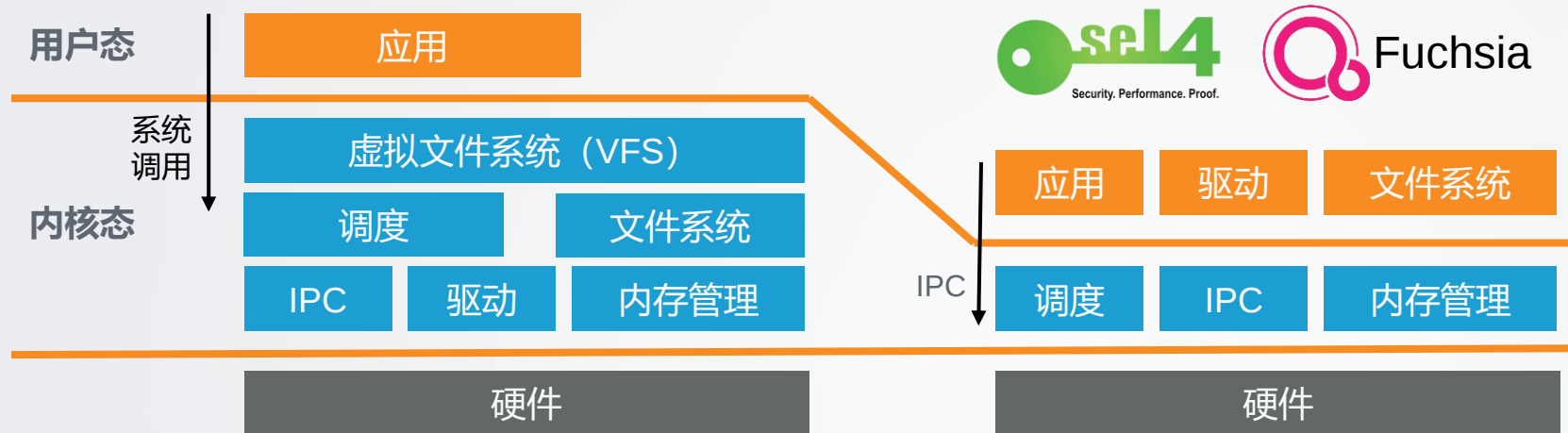
来源：许中兴博士演讲PPT

亿欧智库：微内核与宏内核的性能对比

|        | 宏内核                          | 微内核                            |
|--------|------------------------------|--------------------------------|
| 代表系统   | Linux、FreeBSD、Windows98、DOS等 | QNX、L4、Symbian、Mac OS X、Minix等 |
| 基本概念   | 用户服务和内核服务运行在相同地址空间中          | 用户服务和内核服务运行在相同地址空间中            |
| 常规尺寸   | 比微内核大                        | 比较小                            |
| 可扩展性   | 扩展较困难                        | 容易扩展                           |
| 执行进程速度 | 快                            | 慢                              |
| 安全性    | 单个服务崩溃往往意味着整个系统崩溃            | 单个服务崩溃不影响全局                    |
| 代码开发   | 需要开发的代码量大                    | 需要开发的代码量少                      |

# 宏内核vs微内核，对比发现微内核是IOT OS的发展方向

## 亿欧智库：两种内核的特点及优缺点对比



### 宏内核 (Monolithic kernel)

**特点：**大量功能集中在内核态实现，内核处于特权级各种应用处于非特权级。

#### 优点：

- 具有近30年的发展历史，有统一的社区和生态。

#### 缺点：

- 安全与可靠性不足，模块之间没有很强的隔离机制。
- 实时性支持不足，系统复杂导致无法做最坏情况的时延分析。
- 系统过于庞大阻碍创新，Linux代码行数超2千万行。

### 微内核 (Micro Kernel)

**特点：**将尽可能多的功能移出内核态至用户态，称为服务(Server)

#### 优点：

- 易于扩展移植，大部分模块与底层硬件无关，满足不同终端对操作系统的个性化定制需求。
- 安全可靠，服务之间存在进程粒度隔离，模块化提升操作系统容错能力。

#### 缺点：

- 性能较差，内核的模块交互由函数调用变成了进程间通信，降低了部分效率。
- 生态欠缺，目前尚未有统一的开发者社区。

## IOT需要什么样的操作系统？

### 无缝更新能力

系统更新通过后台完成，无需中断

### 维护系统安全能力

具备防止恶意攻击能力

### 保持长期连接能力

保持 5G、WIFI等连接，保证设备间能一直相互连接

### 云端接入能力

支持设备间无缝访问数据

### 支持各种交互能力

兼顾触控、手写、语音、键鼠等方式以及能够通过传感器和姿势感知

### 支持多样产品形态能力

支持云、边、端的应用

### AI能力

## 微内核是IOT OS的发展方向

# Fuchsia以移动平台为入口，进军桌面平台、服务器平台，充分利用Android生态，拓展在物联网行业的合作伙伴

Fuchsia极有可能是谷歌告别Linux kernel历史包袱，并且建立起统一操作系统的契机。Google挑选了一系列它认为合适的技术和组件加入到这个新操作系统中。例如，微内核Zircon、基于能力的访问控制、Vulkan图形接口、3D桌面渲染Scenic、Flutter应用开发框架等。谷歌对其的发展规划是，先在智能音箱等智能家居上应用，然后逐渐转移到笔记本电脑等更大的设备上。用于编写 Fuchsia 应用的 Flutter SDK 现已经能够为 Android 和 iOS 应用程序生成代码，开发跨系统应用正在尝试中。

亿欧智库：加入Fuchsia生态的产业巨头

| 国家 | 公司                                       |
|----|------------------------------------------|
| 中国 | 终端制造商：华为、小米、OPPO、Vivo<br>半导体：紫光展锐、联发科    |
| 美国 | 半导体：高通                                   |
| 英国 | 图形技术：Imagination Technologies<br>半导体：ARM |
| 日本 | 终端制造商：索尼、夏普                              |
| 韩国 | 终端制造商&半导体：三星                             |
| 印度 | 嵌入式&无线通信：GlobalEdge Software             |
| 瑞士 | 半导体：意法半导体                                |

亿欧智库：Fuchsia在各个平台的潜在优势

服务器平台

- 原生的进程沙箱机制将带来新的安全特性和容器机制

桌面平台

- 游戏3D引擎Pipeline的图形栈，和毫无遗产负担的实现使电子娱乐的实现更为高效；
- 无缝兼容庞大的Android生态，真正实现桌面端和移动端的打通。

移动平台

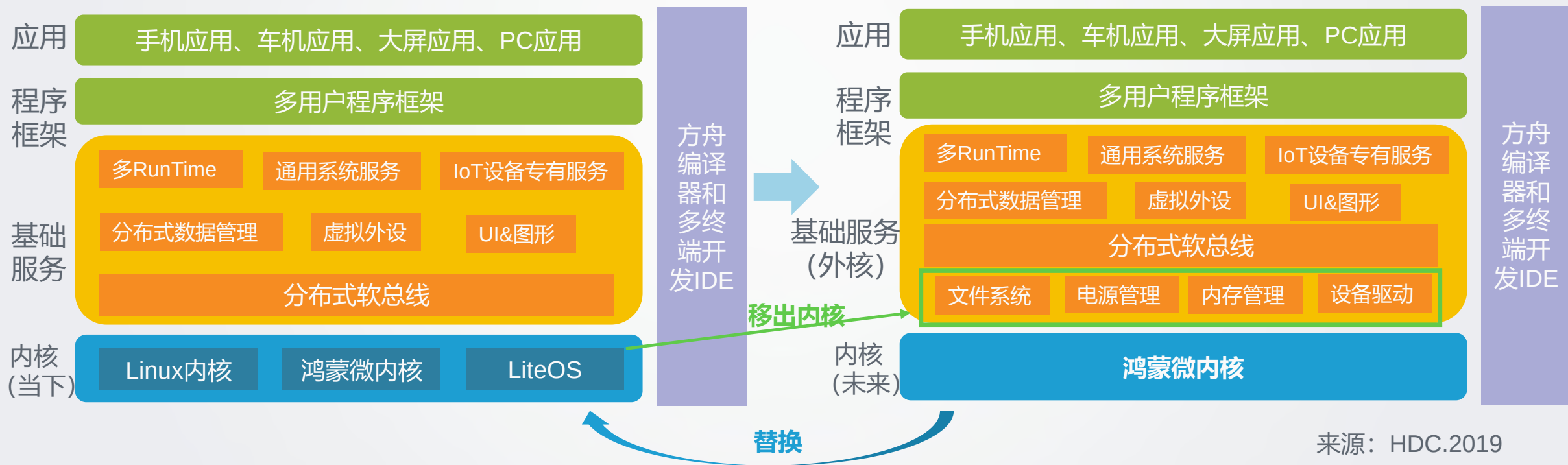
- 系统模块化带给第三方设备厂商全面定制的能力；
- 驱动框架方便硬件厂商编写和维护私有驱动。

来源：许中兴博士演讲PPT

# 华为：物联网操作系统的发展新方向是鸿蒙OS，变革关键点是以鸿蒙微内核取代目前多种内核共同运作的模式

华为对鸿蒙OS的发展给出了大致的发展阶段，其中最核心的转变是在兼容Linux内核的基础上逐渐用自研的鸿蒙微内核替换掉当下Linux内核、鸿蒙微内核和华为LiteOS共存的形态。同时鸿蒙也会关注于对Android的兼容问题。分布式技术的广泛运用为终端设备之间的互操作性提供可能。方舟编译器和面向多终端开发的IDE为代码一次开发多终端移植提供工具便利。

亿欧智库：华为鸿蒙OS架构图（当下和未来）



来源：HDC.2019

**分布式技术**带来的设备间互操作性可能是鸿蒙OS带给消费者的最大不同体验。以分布式软总线、分布式数据管理和分布式安全为三大核心的分布式技术从根本上将其区别于之前的主流操作系统。同时，分布式技术也将改变上层应用和终端硬件的开发形态。

## 分布式软总线

分布式软总线是对过往连接中硬总线的替代，全场景设备间可以基于软总线完成设备虚拟化、跨设备服务调用、多屏协同、文件分享等分布式业务，具备自发现、自组网、高带宽低时延的特点。

**低时延** 10ms端到端时延

**高吞吐** 有效吞吐2.4Gbps

**高可靠** 30%抗丢包率

**异构融合网络** WiFi、蓝牙等异构网络快速连接

**自发现自组网** 无感连接

**动态时延校准** 多设备间的时延同步

## 分布式数据管理

分布式数据管理让跨设备处理如同本地一样方便快捷，本质上是不同设备上的数据整理成一个可以统一访问的接口去调用。

**分布式文件系统** 远程读写性能是Samba的4倍

**分布式数据库** OPS性能是ContentProvider的1.3倍

**分布式检索** 检索性能是IOS Core Spotlight的1.2倍

## 分布式安全

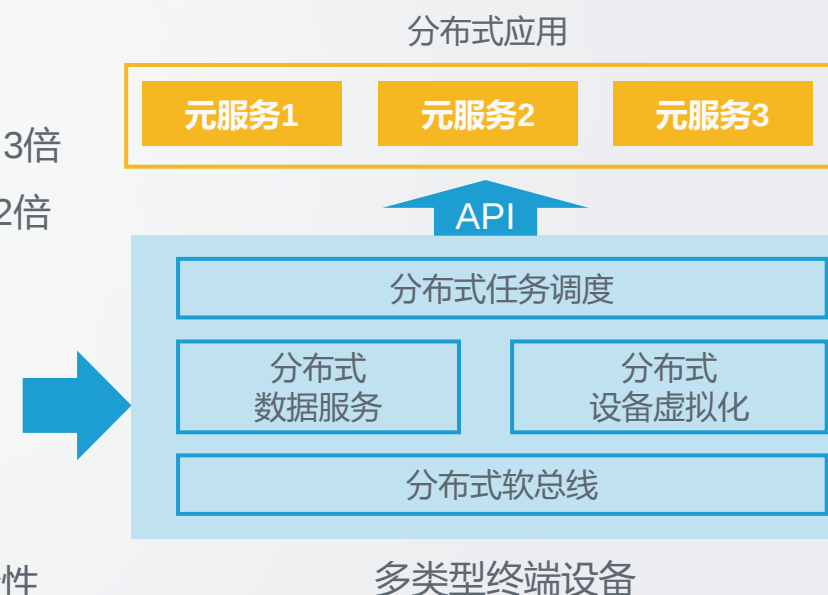
确保正确的人、用正确的设备、正确的使用数据

**正确的人** 根据业务场景提供最佳安全认证策略

**正确的设备** 内核级安全能力提升全场景设备安全性

**正确使用数据** 定义数据和设备安全级别确保数据流通安全可靠

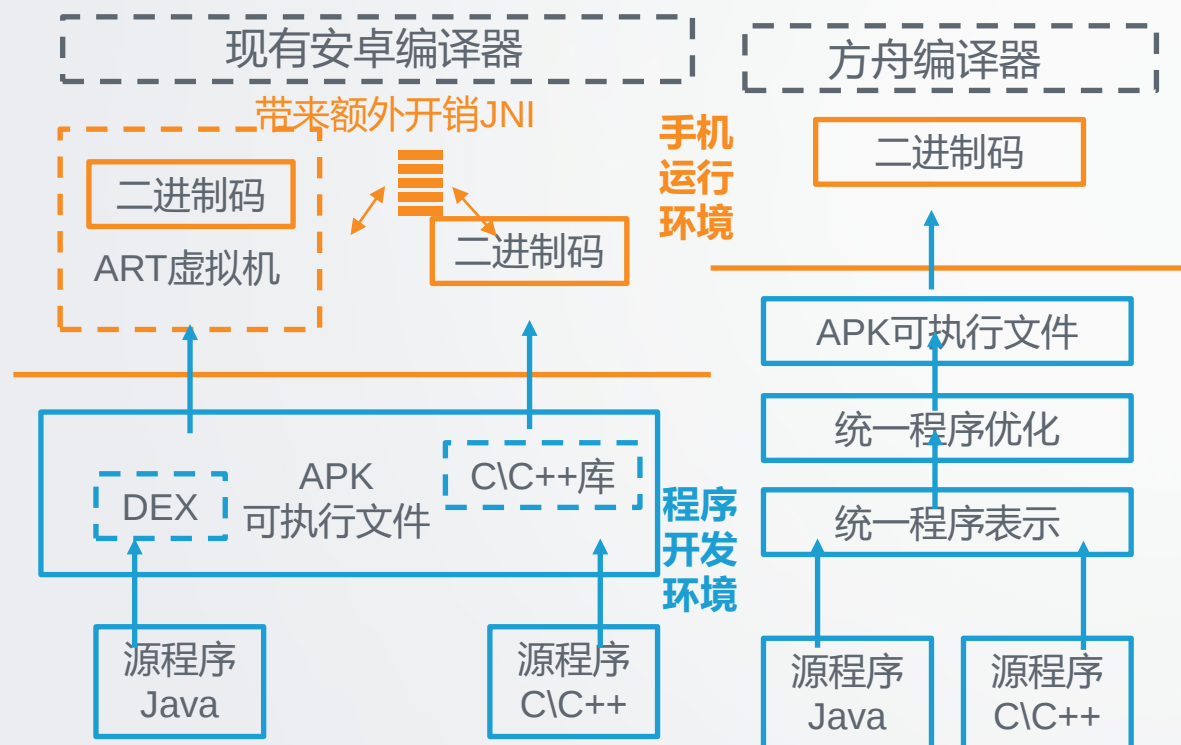
系列分布式技术带来应用层面的形态改变，一个应用由许多可以独立运行的**元服务**组成，以服务分布式条件下任务调用的需求。



# 方舟编译器对终端软件编译机制进行了架构级优化，在鸿蒙OS应用开发中起关键作用

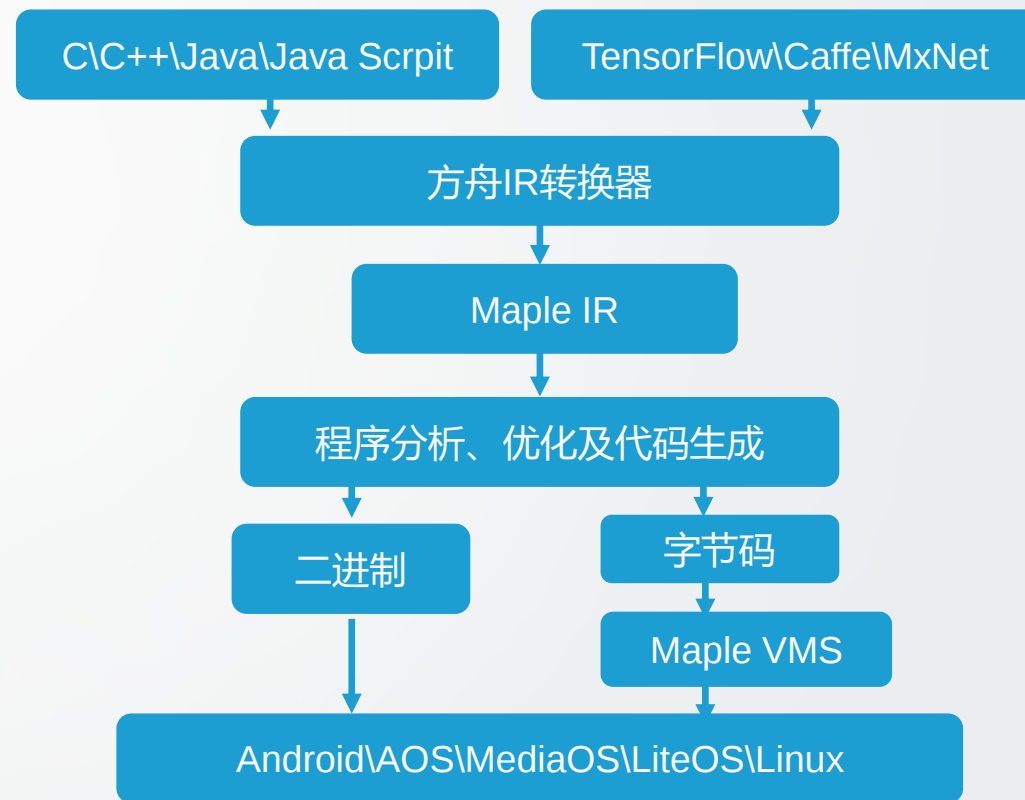
方舟编译器将会是鸿蒙OS赢得开发者生态的关键。跨语言编译是制约应用程序执行效率的瓶颈，从程序员使用的C、C++和Java等程序语言转换为硬件执行的机器码。和现有安卓编译器不同代码保持独立，在运行环境中进行协调，因此产生额外系统开销的做法不同，方舟编译器在开发环境中就编译成一套可执行文件，节约了系统开销。多层Maple IR能够使得多种编程语言在编译后可兼容运行于不同操作系统下。

## 亿欧智库：方舟编译器对开发环境的精简



来源：2019.04发布会

## 亿欧智库：Maple IR是方舟编译器的核心



来源：史宁宁，Maple IR分析

# 鸿蒙OS的发展将紧密结合华为 “1+8+N” 的物联网终端战略

如前文所述，华为Lite OS在轻量级物联网操作系统上早有布局，而2019年8月发布的鸿蒙OS则是更进一步的升级。作为对比，我们可以发现鸿蒙OS与Fuchsia在设计理念上极为相似。鸿蒙OS也采用微内核架构，将系统分为小的功能模块，只在内核中保留最核心的调度等功能。可以预见未来鸿蒙OS将会通过华为 “1+8+N” 的终端战略搭载在各种不同形态的产品上。

亿欧智库：华为 “1+8+N” 终端战略

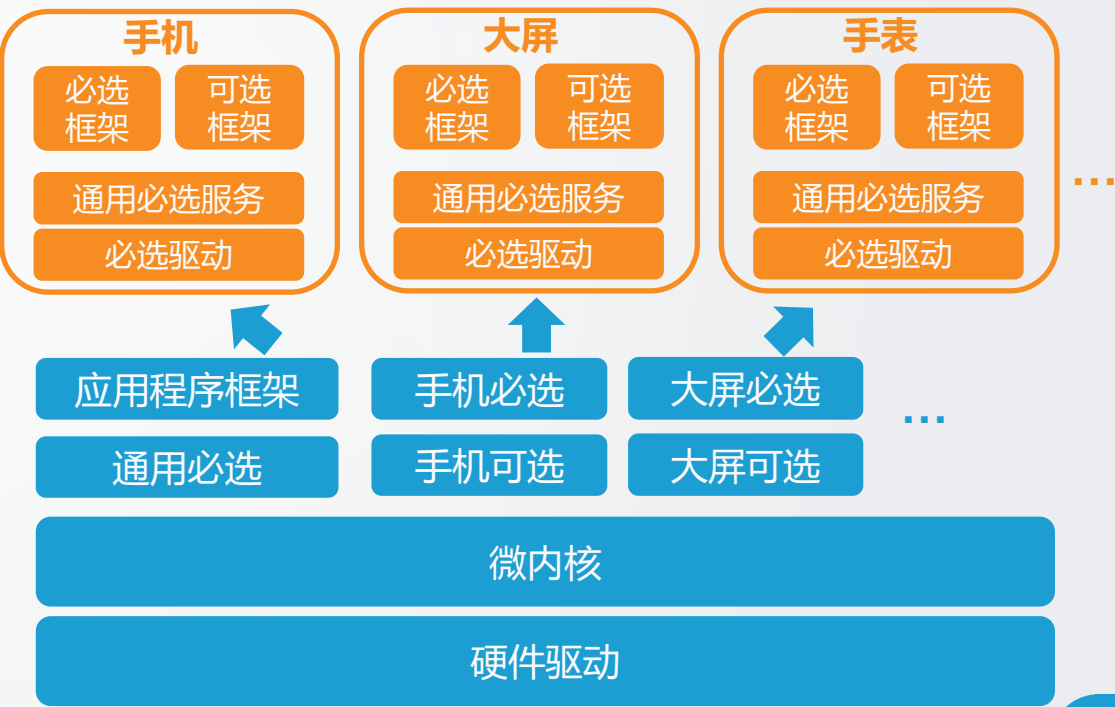


**1+8+N:** 1个主入口（手机），8个辅入口（平板、PC、车机、HD、音箱、耳机、VR、穿戴）  
N指广泛的IoT设备（生态伙伴设备）



**可裁剪:** 系统功能可按照不同终端需求实现个性化定制。  
**跨终端:** 综合信息感知平台实现多终端能力共享。  
**易开发:** 一次开发，多端部署；控件标签化，布局自适应。

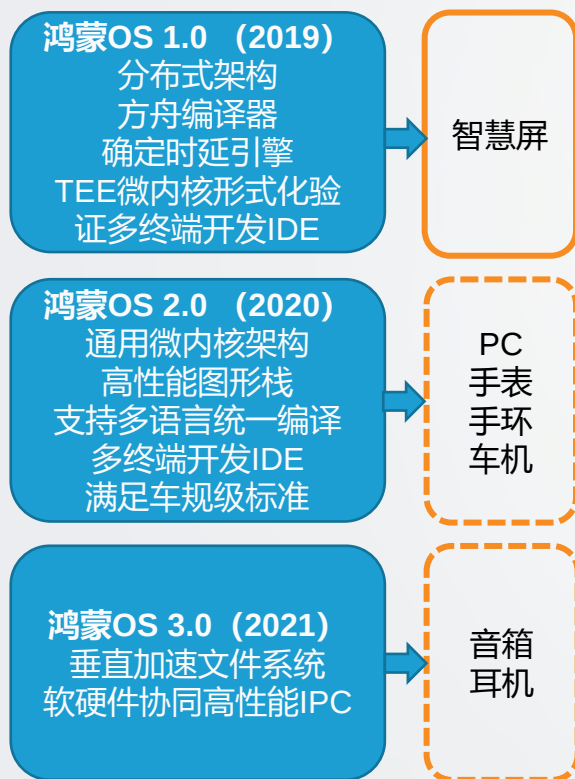
亿欧智库：可裁剪特性是鸿蒙OS的主要特点



# 华为鸿蒙OS的产品落地情况

华为消费者业务坚持“1+8+N”全场景智慧生活战略，以鸿蒙操作系统和HiAI为核心驱动力，支持HiLink智能家居生态和HMS（华为终端云服务, Huawei Mobile Services）服务生态的协同创新，升级软硬件用户体验，使包括手机在内的全场景终端实现互联互通、能力共享，让消费者享受不同生活场景下、多种设备间无感连接、信息内容无缝流转的智慧生活美好体验。

## 鸿蒙OS未来近三年发展规划



## 鸿蒙OS首次应用——智慧屏

鸿蒙OS+AI摄像头赋能传统电视，实现智慧交互 多屏协同 IoT控制 影音娱乐

**华为智慧屏 V65、V75、V55i、V65i**  
操作系统：Harmony OS 1.0

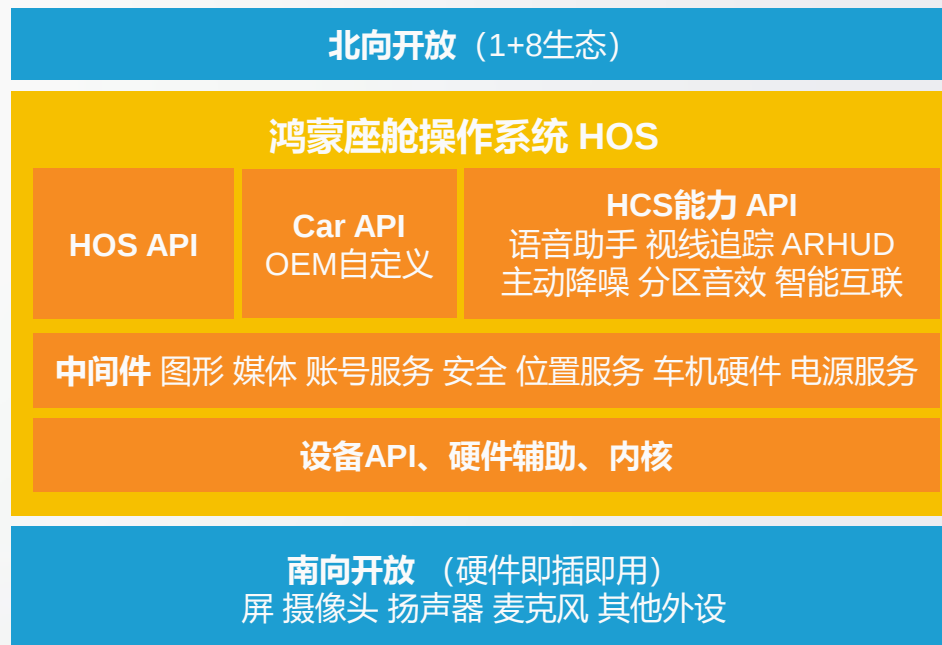
- AI健身 人性互动
- WeLink 多方视频通话
- HUAWEI Share 图片互传
- Magic-link 一碰投屏，超低时延

**华为智慧屏 X65**  
操作系统：Harmony OS 1.1  
AI健身2.0 AI KIDS2.0 AI智能回看

**华为企业智慧屏 Idea Hub**  
操作系统：Harmony OS 1.1  
智能音幕 开放软硬件，易集成

## 鸿蒙OS发展新方向——软件定义汽车

鸿蒙座舱操作系统HOS：软硬件解耦，南北向开放，可实现语音交互、视觉识别、音频优化，预计2020年落地



## 四、物联网操作系统发展展望

The outlook for the development of IoT OS

# AIoT时代的操作系统发展路径很有可能区别于桌面和移动时代

尽管桌面和移动时代操作系统的发展给我们提供了很多信息和经验，但AIoT时代的操作系统发展路径很有可能区别于桌面和移动端的操作系统。相比于设备硬件高度相似、应用通用性高的桌面端和移动端，IoT OS面临的场景更为复杂，设备种类更多，因此也需要更强的协同作业能力。

## 亿欧智库：通用型计算机系统和物联网系统的对比

| 特征    | 通用型计算机系统                                | 物联网系统                                     |
|-------|-----------------------------------------|-------------------------------------------|
| 形式和类型 | 看得见的计算机形式；按体系结构、运算速度和结构规模，可分为大、中、小、微型机。 | 看不见的计算机；形式多样，应用领域广泛，按应用场景来分类。             |
| 组成    | 通用处理器、标准总线 and 外设；软件与硬件相对独立。            | 面向应用的嵌入式处理器，总线和外部接口多集成于处理器内部；软件与硬件集成紧密。   |
| 开发方式  | 开发平台和运行平台都是通用计算机。                       | 采用交叉开发方式，开发平台为通用计算机（宿主机），运行平台为类嵌入式系统（目标机） |
| 二次开发  | 应用程序可重新编制                               | 一般不能再编程，或二次编程较复杂                          |

## 亿欧智库：以嵌入式系统为基础的物联网系统发展趋势

**趋势1：**嵌入式系统以行业应用为主，不会像桌面和移动端一样存在单一或寡头平台；而同一行业内有可能会形成开放、软硬件重用、构件兼容、维护方便和产业分工的标准，以增强行业内的健康发展。

**趋势2：**嵌入式应用开发急需强大的开发工具和操作系统的支持，降低开发难度，提升开发速度。

**趋势3：**嵌入式系统逐渐向可编程的新型计算模型发展。

**趋势4：**在不同应用行业领域，收取开发费用和版税的商用型OS和不收版税，只收技术服务费的开源型OS会共同存在，为建立以生态为主的行业巨头尤其会选择开源模式。

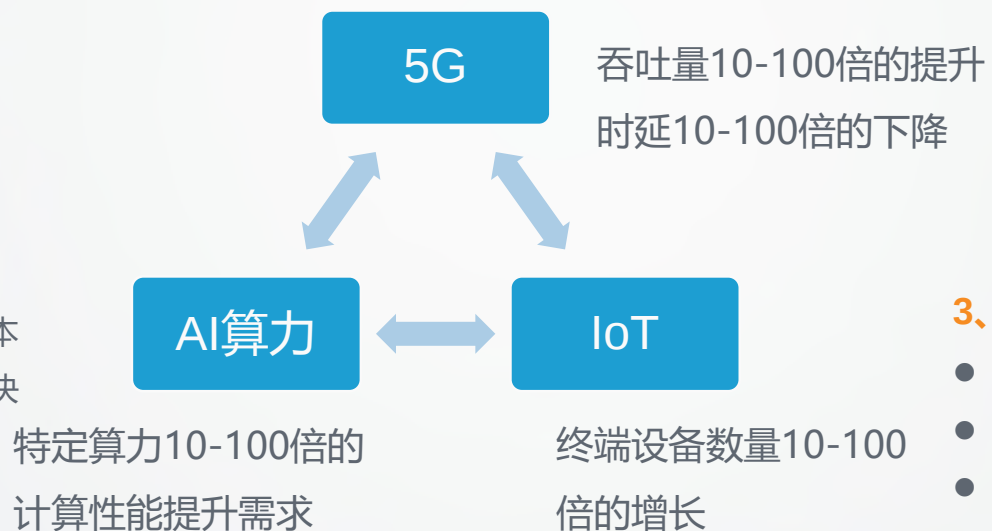
# 现在和将来：AIoT时代，操作系统层面的角逐刚刚开始

AIoT时代，软硬件的需求都发生变化，对介于软硬件之间的操作系统层产生新需求。我们相信AIoT时代必将带来操作系统层的新机会。从软件层面，IoT OS 拥抱AI是必然，物联网在异构数据收集上天然具有优势，但AI在物联网的整体融入则在算力和算法层面提出不小挑战。同时，5G等通信技术带来的高吞吐和低时延对OS性能也提出了新的要求。AIoT是人-机-物高度融合的物理信息系统（CPS,Cyber-Physical System），操作系统层面面临挑战，这需要科技企业和开发者去推动生态和技术演化之路。

## 亿欧智库：5G+AIoT对物联网操作系统提出的新要求

### 2、AI对OS形成的挑战：

- OS如何统一调度不同类型的芯片加速器（MCU\FPGA\SoC）以实现目的
- 碎片化场景下，大部分终端设备的成本和算力无法满足AI需求的问题如何解决



### 1、低时延要求对OS形成的挑战：

- 操作系统成为制约整体时延的关键因素；
- OS对5G低时延的适应与进化；
- OS在数据高吞吐量下能否实现工作的稳定性。

### 3、海量终端对OS形成的挑战：

- 海量设备的差异化需求如何由软件来实现定义和满足
- 多类型设备的互操作和协同如何实现
- 连接设备资源从KB级到TB级的跨度；
- 开放的IoT网络如何确保海量设备的安全

IoT OS向大型（复杂可配置）和小型（安全和硬化）两个方向进行同步发展

# 亿欧 AIOT时代的新军备：操作系统应该怎么做？

从第二章我们发现，无论是硬件厂商、云计算公司还是传统软件开发商，都在积极备战AIoT技术储备。**我们认为操作系统是一家致力于在AIoT领域起到关键性作用和占据重要产业地位的公司所必须掌握的核心技术之一。**但目前不管从技术成熟度还是应用成熟度而言，面向AIoT的操作系统发展任重道远，由此，我们提出了以下几个重点问题和行业内进行探讨：

## 1、企业开发的面向物联网的操作系统究竟应该开源还是闭源？

- 开源的优势：**替软件开发者缩减成本，迅速积累开发者的使用和支持；  
替硬件开发者降低系统适配难度；  
后续开发中可以借助外部开发者的力量。
- 劣势：**对软件缺乏掌控力，容易出现碎片化的版本失控情况；  
软件本身商业变现存疑，商业模式待探索；
- 闭源的优势：**加强企业对软件的控制，标准化好，能够从软件本身获得利润；
- 劣势：**一般以牺牲市场占有率为代价，发展速度慢；  
对企业的后续服务能力提出考验，需要持续投入自有资源到软件升级优化和开发者套件的开发中。
- IoT OS的发展中是否会出现类似于谷歌AOSP（Android开源项目）+GMS（谷歌移动服务）的控制模式值得思考。当然很有可能会产生软硬件厂商更加紧密合作发展的新模式。

## 2、面向物联网的操作系统应该是统一的还是多套系统共存？

物联网终端的碎片化和复杂度使得是否可能存在统一的一套系统值得商榷。此前并非没有厂商尝试过一统多终端的操作系统，最典型的案例是微软曾尝试用“OneCore”项目将Windows 10，Windows Server，Windows 10 Mobile和Windows 10 IoT等构建在同一Core之上。用一套OS内核实现对终端设备的全覆盖，目前来看是一项极难的挑战。作为对比，目前同一厂商基本会开发几套系统：

|           | 小资源场景          | 中等资源场景           | 资源较丰富场景                 |
|-----------|----------------|------------------|-------------------------|
| 微软        | Azure RTOS     | Win10 IoT Core   | Win10 IoT for indyustry |
| 阿里云       | AliOS Things   | AliOS            |                         |
| 华为        | HuaWei LiteOS  | Harmony OS       |                         |
| 谷歌        | Fuchsia        |                  |                         |
|           | Wear OS        | Android Thing    | Android Auto            |
| RT-Thread | RT-Thread Nano | RT-Thread IoT OS | RT-Thread Smart混合内核     |

注：此处仅考虑物联网相关OS

不仅是开源\闭源，统一操作系统\多种OS共存的发展问题，限制物联网产业发展的核心因素之一，难以像手机端一样建立硬件和软件开发解耦合的虚拟抽象层的问题一直无法解决。

### 3、物联网产业如何应该实现硬件标准和软件开发的分离？

以安卓为例，在建立了硬件标准接口和用虚拟抽象层实现软硬件分离与解耦后，才使得软件开发者只需调用OS提供的编程接口，而无需关心硬件配置。

而显然在物联网中开发这样一层物理世界的抽象层难度远远大于手机这一单一设备。不同厂商之间从建立自身生态考虑出发，也给物联网设备间的互联互通带来阻碍。物联网统一标准的道路上还存在着大量“解耦”工作，需要通过不同层面的操作系统，将硬件相对标准化，定制化功能通过软件实现。

软件定义物联网，可能还要很长的路要走。

### 4、在IoT OS发展及其生态上，国内是否可以借机实现对国外的赶超？

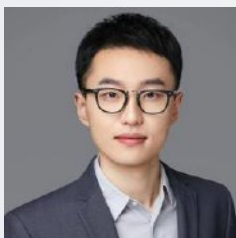
自1980s以来的嵌入式系统发展过程中，国内伺服器、工控场景对RTOS的开发需求缺失直接导致我国在嵌入式操作系统发展上落后于国外，国内基本采用的是国外的FreeRTOS、VxWorks或嵌入式Linux等。更不要谈Windows、Android、iOS等国外研发的通用型操作系统在国内的垄断地位。

但随着近年来我国对物联网产业发展的实际需求产生和重视，涌现了一批优秀的IoT OS厂商和相关生态。可以说在探索物联网的新系统和新应用上，我国很难再从国外发达国家处获得有效借鉴。

由于软硬件生态原因，对国外成熟操作系统厂商（例如微软和谷歌）的产品进行追赶极难实现，但在物联网的发展上，国内厂商及其指定的标准可能有机会实现弯道超车。

- 经过桌面研究及对相关企业、行业专家访谈后完成此份报告。在前序报告的基础上，围绕物联网产业中相当重要的端侧操作系统展开研究。在梳理、分析和比较了目前市面上应用于物联网场景较多的四类物联网操作系统后，提出我们对物联网产业中操作系统的看法。需要注意的是，因为物联网操作系统的发展还有许多不确定性，因此我们更多抱有和行业内专家及从业者探讨的态度。
- 未来亿欧智库将持续追踪、研究物联网领域，进行更深入探讨，持续输出更多研究成果。欢迎大家与我们联系交流，提出宝贵意见。

## ◆ 报告作者



朱涛

分析师

Email: zhutao@iyiou.com



孙舒展

实习分析师

Email: sunshuzhan@iyiou.com

## ◆ 报告审核



李星宏

高级分析师

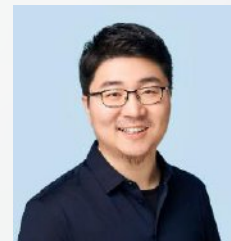
Email: lixinghong@iyiou.com



石晓霞

研究副总监

Email: shixiaoxia@iyiou.com



由天宇

亿欧公司副总裁、亿欧智库院长

Email: youtianyu@iyiou.com

## ◆ 团队介绍:

- 亿欧智库是亿欧公司旗下专业的研究与咨询业务部门。
- 智库专注于以人工智能、大数据、移动互联网为代表的前瞻性科技研究；以及前瞻性科技与不同领域传统产业结合、实现产业升级的研究，涉及行业包括汽车、金融、家居、医疗、教育、消费品、安防等等；智库将力求基于对科技的深入理解和对行业的深刻洞察，输出具有影响力和专业度的行业研究报告、提供具有针对性的企业定制化研究和咨询服务。
- 智库团队成员来自于知名研究公司、大集团战略研究部、科技媒体等，是一支具有深度思考分析能力、专业的领域知识、丰富行业人脉资源的优秀分析师团队。

## ◆ 版权声明:

- 本报告所采用的数据均来自合规渠道，分析逻辑基于智库的专业理解，清晰准确地反映了作者的研究观点。本报告仅在相关法律许可的情况下发放，并仅为提供信息而发放，概不构成任何广告。在任何情况下，本报告中的信息或所表述的意见均不构成对任何人的投资建议。本报告的信息来源于已公开的资料，亿欧智库对该等信息的准确性、完整性或可靠性作尽可能的追求但不作任何保证。本报告所载的资料、意见及推测仅反映亿欧智库于发布本报告当日之前的判断，在不同时期，亿欧智库可发出与本报告所载资料、意见及推测不一致的报告。亿欧智库不保证本报告所含信息保持在最新状态。同时，亿欧智库对本报告所含信息可在不发出通知的情形下做出修改，读者可自行关注相应的更新或修改。
- 本报告版权归属于亿欧智库，欢迎因研究需要引用本报告内容，引用时需注明出处为“亿欧智库”。对于未注明来源的引用、盗用、篡改以及其他侵犯亿欧智库著作权的商业行为，亿欧智库将保留追究其法律责任的权利。



网址: <https://www.iyiou.com/research>

邮箱: [zk@iyiou.com](mailto:zk@iyiou.com)

电话: 010-57293241

地址: 北京市朝阳区霞光里9号中电发展大厦A座10层