arm

The Right Managed OS To Power A Trillion Connected IoT Devices

FEATURING RESEARCH FROM FORRESTER

Leverage IoT Device OS To Boost Digital Transformation

The days of customized RTOS work are long gone; welcome to the era of powerful, nimble operating systems that reduce development time and complexity

The opportunity to exploit the vast new world of the Internet of Things (IoT), to use data and its insights to transform your business, is knocking at your door. Whether it's reducing costs and increasing efficiency, improving customer experience, developing new revenue streams, or some other driver, you have an opportunity to leverage the IoT in ways that have never been easier, more accessible or broadly supported.

But the opportunity to take advantage of this technological transformation may not appear easy. There are a lot of moving parts in developing an IoT solution, and traditional ways of approaching such development don't scale. For example, the vast majority of companies maintain their own operating systems to run their embedded devices, customized for their applications. However, IoT changes the equation for embedded, with these systems now demanding internet connectivity over different communication channels. With that comes security and manageability considerations.

This complexity can disrupt existing development practices, strain team capacity and ability, and balloon your budget. The days of roll-your-own embedded operating systems powering a company's devices and solutions are gone. It's just not an economic or scalable way to design IoT. But this has opened up the demand for a new solution: the IoT operating system.

A NEW APPROACH TO IOT DEVICES

So how and what to choose? This Forrester report offers a comprehensive look at the embedded operating system choices developers have before them today, the strategic benefits, and the typical architecture of an IoT device OS. Additionally, it defines market segments and presents representative open source and commercial offerings in each segment.

Arm's Mbed OS is listed among them. A free, open source, RTOS-based IoT Platform for Cortex-M embedded devices, Mbed OS is integrated with all the fundamental components that any successful operating system needs including

- Connectivity such as BLE, WiFi, cellular, NB-IOT
- Security, including channel and device security
- Services for managing and updating devices.

IN THIS DOCUMENT

All this also comes supported by a range of professional tools, a diverse IoT ecosystem and the option to purchase commercial support.

Mbed gives developers the resources, knowledge, know-how and connections to diligently, methodically and meticulously explore IoT right from the start. Mbed has grown from the earliest days of IoT, and savvy professional developers have used the OS to orchestrate and shape the foundations of IoT innovation.

It's built on the following core principles:

- **Experience and know-how:** Mbed was launched 15 years ago for embedded designs, built on the back of decades of innovation that Arm delivered into the mobile space.
- **Innovation collaboration:** The Arm ecosystem for IoT has more than 360,000 developers and more than 80 silicon, module, board and tools partners relentlessly working in concert with Arm technology experts to deliver the most innovative solutions into the IoT market.
- **Continuous investment:** Arm and the Arm ecosystem continually invest in development, testing and debugging tools while leveraging Arm hardware IoT innovation to stay ahead of the design curve. Commercial support for Mbed OS is available for customers requiring faster response times, access to Mbed IoT experts, and prioritized effort on critical issues, enabled through a technical account manager and private support portal.
- **Scaling and future-proofing:** Leveraging Arm's experience in mobile and IoT, Arm Mbed OS is updated quarterly, and backed by maintenance updates to ensure that system designs are flexible enough to last decades.
- **Foundation for IoT:** Arm Mbed OS is also designed to work seamlessly with Arm Pelion IoT, the first horizontal platform that provides integrated connectivity, security, management, and drivers needed for scalable IoT systems.

MORE EFFICIENT AND EFFECTIVE DESIGN

Mbed OS accelerates product development by 90% by integrating all the necessary connectivity, security and software components, across many hardware solutions. This frees businesses to focus on building unique product applications and getting to market faster.

It also implements the Arm Platform Security Architecture (PSA) to provide device security and uses Mbed TLS to provide industry-leading transport security.

The Mbed OS platform is built to be portable across different hardware, enabling the same application code to run across different devices. This enables developers to select the right technology and vendor for a specific device. Using the same software platform and tools across multiple products that may use different devices and vendors, and even allow for multi-sourcing.

All these features take much of the formerly ground-up development work off the shoulders of design teams to free them to do what they do best: Create and optimize the best IoT solutions for their customers.

Take for example a generic asset-tracking device. Only the RTOS kernel, timers, and power modes are generally available with a traditional RTOS. But Mbed OS includes that plus cloud libraries, connectivity modules, peripheral drivers, crypto and file system functionality and more. The latest release, Mbed OS 5.10, enhances the user experience for future over-the-air upgrades, support for Armv8-M and Cordio BLE 5 stack open sourcing.

WHAT LIES AHEAD

This year, developers will see new Mbed features. These include more integrated PSA features along with PSA compliance and certification, open source Cordio BLE stack running on a number of partner platforms, release of Mbed Studio desktop 1.0, open source BLE mesh implementation on Mbed OS, Wi-SUN, the launch of Mbed OS 6.0 which is set to influence a step change in operating systems and Mbed Linux OS.

Additionally, the Mbed Linux OS extends Mbed OS to the products based on Arm's Cortex-A series processors. The Mbed team has taken the Linux kernel and popular Linux tools and used these to build a free, open source operating system that meets the needs of IoT devices. For example, in the same way Google uses Linux in its Android smartphone OS, Arm uses Linux to build a leading OS for smart IoT devices.

CONCLUSION

A managed real-time operating system is vital for many types of applications. Care must be taken in resource-constrained, power-sensitive applications such as IoT devices to choose the right OS to ease development and ensure market success. The goal to reach one trillion connected devices by 2035 means the process of conceiving, developing and deploying IoT devices and solutions has to accelerate. Also, as time-to-market pressures mount and solution complexity soars, there is little need to task internal development teams with building one, especially when choice abounds.

Customers realize this already: A Forrester Opportunity Snapshot released earlier this year found that the most critical capability that customers look for in their development partners is real-time capabilities followed closely by the ability to scale.

By listening to customers and partners and innovating relentlessly, Arm has built a powerful and nimble RTOS solution and ecosystem that has powered countless transformative IoT solutions around the world. We'd love for you to join the movement. A third of a million developers can't be wrong.

Note: The next version of Mbed OS (5.12) will be launched in March 2019.

FORRESTER®

Leverage IoT Device OS To Boost Digital Transformation

Choose An Internet Of Things Device Operating System To Bridge Physical Operations And Digital Businesses

by Charlie Dai

October 17, 2018

Why Read This Report

The operating system (OS) running on a wide range of internet of things (IoT) devices, such as sensors, home appliances, industrial gateways, and edge devices, is a fundamental building block of the IoT software stack. To accelerate their firms' digital transformation, enterprise architecture (EA) pros must have a holistic view of IoT device OSes. This report analyzes the strategic benefits and typical architecture of an IoT device OS, defines market segments, and presents representative open source and commercial offerings in each segment for architectural consideration.

Key Takeaways

Data Challenges Drive A Focus On IoT Device Operating Systems
Integrating data streams from IoT devices with enterprise systems poses significant challenges. The IoT device OS is critical to addressing these challenges and generating reliable insights from these connected devices.

IoT Device OS Architecture Is Key To Enabling IoT Development And Operations (DevOps)
IoT device OS software has five architecture layers: kernel, hardware abstraction, IoT enablement, application, and security. EA pros must map the DNA of IoT device OSes to their application scenarios.

Three Categories Of IoT Device OS Are Critical For Implementation
Most IoT device OSes are open source. EA pros should have a holistic view of these OSes and choose among IoT device OS frameworks, full-stack IoT device OSes, and embedded device OSes.

This PDF is only licensed for individual use when downloaded from forrester.com or reprints.forrester.com. All other distribution prohibited.

Leverage IoT Device OS To Boost Digital Transformation

Choose An Internet Of Things Device Operating System To Bridge Physical Operations And Digital Businesses

by Charlie Dai
with Glenn O'Donnell, Frederic Giron, Michele Pelino, Han Bao, and Bill Nagel
October 17, 2018

Table Of Contents

Related Research Documents

Boost Digital Business With The Internet Of Things

Design Your Integration Architecture For The Internet Of Things

TechRadar™: Internet Of Things, Q1 2016

Use IoT To Link Physical Operations And Digital Business

Share reports with colleagues.
Enhance your membership with Research Share.

FORRESTER®

Forrester Research, Inc., 60 Acorn Park Drive, Cambridge, MA 02140 USA
+1 617-613-6000 | Fax: +1 617-613-5000 | forrester.com

© 2018 Forrester Research, Inc. Opinions reflect judgment at the time and are subject to change. Forrester®, Technographics®, Forrester Wave, TechRadar, and Total Economic Impact are trademarks of Forrester Research, Inc. All other trademarks are the property of their respective companies. Unauthorized copying or distributing is a violation of copyright law. Citations@forrester.com or +1 866-367-7378

IoT Device OS Deserves Strategic Attention On Your Digital Journey

The internet of things creates data about the physical world and provides new instruments that all companies can use to boost digital business.[1] The fast-growing range of network connectivity options and rapid sensor innovation enable enterprises worldwide to use IoT sensors and actuators to optimize processes and asset utilization as well as differentiate their products and services.[2] In Forrester's IoT software stack architecture, the IoT device operating system is a key component that spans multiple layers and deserves EA pros' attention.[3] Forrester defines an IoT device OS as:

> *A lightweight operating system that runs on diverse IoT devices, manages device hardware and software resources, and provides common services and APIs for IoT connectivity, management, security, computing, and analytics.*

DATA INTEGRATION CHALLENGES DRIVE A FOCUS ON THE DEVICE OPERATING SYSTEM

As more and more devices connect to the enterprise network, EA pros realize that embedding IoT capabilities within products, services, or enterprise business processes employs a wide variety of integration and connectivity patterns. Almost half of global data and analytics decision makers that Forrester surveyed found that integrating enterprise data with IoT data is one of their firm's biggest challenges to generating insights using IoT technologies (see Figure 1). Why? Comprehensive IoT software platforms fall short of reliably addressing these challenges due to heterogeneity and complexity. That's where the IoT device OS can help, because it:

› **Enables heterogeneous devices via hardware abstraction.** The hardware environments that generate IoT data are extremely diverse, including CPUs and GPUs like ARM or x86-based processors; peripherals such as digital cameras, smart speakers, and PCI buses; and hundreds of types of sensors, such as thermocouples, speed sensors, and gas sensors. The IoT device OS abstracts the complexity of device hardware, ensuring the application's compatibility in the meantime.

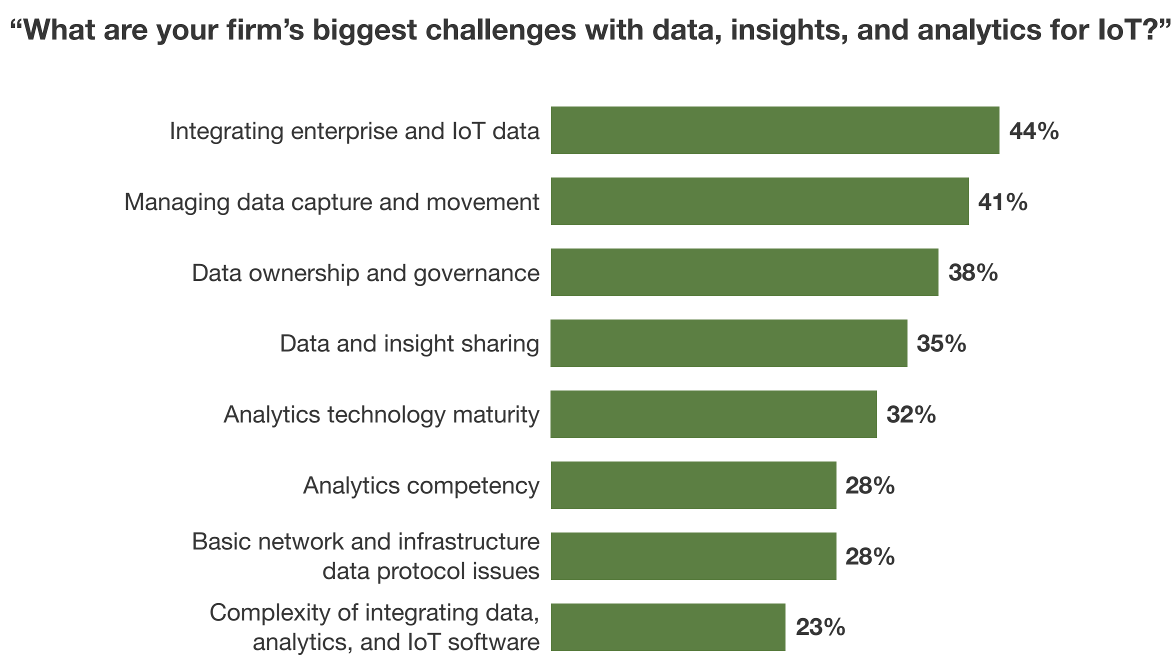
› **Captures data from IoT devices as the source of insights.** IoT device OSes allow IoT devices to capture data in different ways. For time-critical device application scenarios, such as sensors in power plants or high-speed trains, real-time operating system (RTOS) IoT device software can capture data in less than a millisecond. For less time-critical cases, such as supporting a richer UI for production line visibility or enhancing customer experiences for a smart home, Linux-based IoT device OSes can ingest, buffer, and preprocess data.

› **Meets different transport connectivity needs.** To transmit data from devices to edge and cloud data centers and drive actions through data processing and analysis, IoT devices must support a wide range of network connectivity.[4] IoT device OSes enable different connectivity protocols for both the transport and application layers, creating a data transportation foundation for complex business scenarios. An OS can support protocols in the licensed mobile spectrum like NB-IoT for broad geographic coverage of smart transportation as well as white-space spectrum for the low-power scenarios of industrial networks.

© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

› **Makes application data transfer from devices reliable.** Security and power consumption are key factors affecting the reliability of device data transfer. First, IoT threats are becoming more sophisticated and effective.[5] Well-designed IoT device OSes can harden the security protection in dimensions such as the runtime execution environment and data storage and transmission. Second, IoT devices vary dramatically in their power consumption.[6] The power management capabilities of IoT device OSes are critical to ensuring that battery-powered devices can send data over the network continuously for eight to 10 years.

FIGURE 1 Data Integration Is The Biggest Challenge For Enterprise Adoption Of The Internet Of Things



Base: 1,958 data and analytics decision makers whose firm uses IoT services

Source: Forrester Analytics Global Business Technographics® Data And Analytics Survey, 2018

Accelerate Digital Transformation With The Right IoT Device OS

Using the IoT device OS to accelerate digital transformation requires far more than just connecting your IoT software platform to IoT devices. EA, infrastructure and operations, and application development and delivery pros should not rely solely on integrated hardware solutions from IoT vendors. They must focus on three areas to guide their IoT device OS strategy: Understanding the functional architecture

© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

of IoT device OSes to facilitate application development and operations; mapping various IoT device OSes to the technical requirements of major IoT application scenarios to address contextual business needs; and gaining a holistic view of the characteristics of different offerings to make the right decision.

UNDERSTAND THE ARCHITECTURE OF IOT DEVICE OS TO FACILITATE DEVELOPMENT AND OPERATIONS

To accelerate application development and digital operations for IoT initiatives, technology leaders must understand IoT device OS architecture. Forrester identifies five layers in the IoT device OS software stack, each with unique functional capabilities and specific business value (see Figure 2):

› **The kernel layer mediates access to device resources.** This layer has five components. Memory management manages physical or virtual access to memory, such as memory allocation and deallocation, memory protection, and memory access control. Scheduling uses algorithms to decide which task should execute at any given time, such as preemptive scheduling and round-robin scheduling. Multithreading allows one physical processor core to run multiple execution threads simultaneously for better processor utilization.[7] Interrupt management detects hardware events and handles processing and response. Power management minimizes power consumption with approaches such as the idle task hook and tickless mode that Amazon FreeRTOS uses for low-power applications.[8]

› **The hardware abstraction layer simplifies access to heterogeneous hardware.** This layer has five components. Sensor drivers include software interfaces for all kinds of sensors and microcontrollers. Peripheral drivers support diverse ancillary devices, such as UART, SPI, and I2C.[9] Processor drivers support x86 and ARM CPUs and GPUs. File system device drivers support different file systems, such as FAT and USB. A low-level API layer provides consistent APIs for developers to communicate with all kinds of board devices.

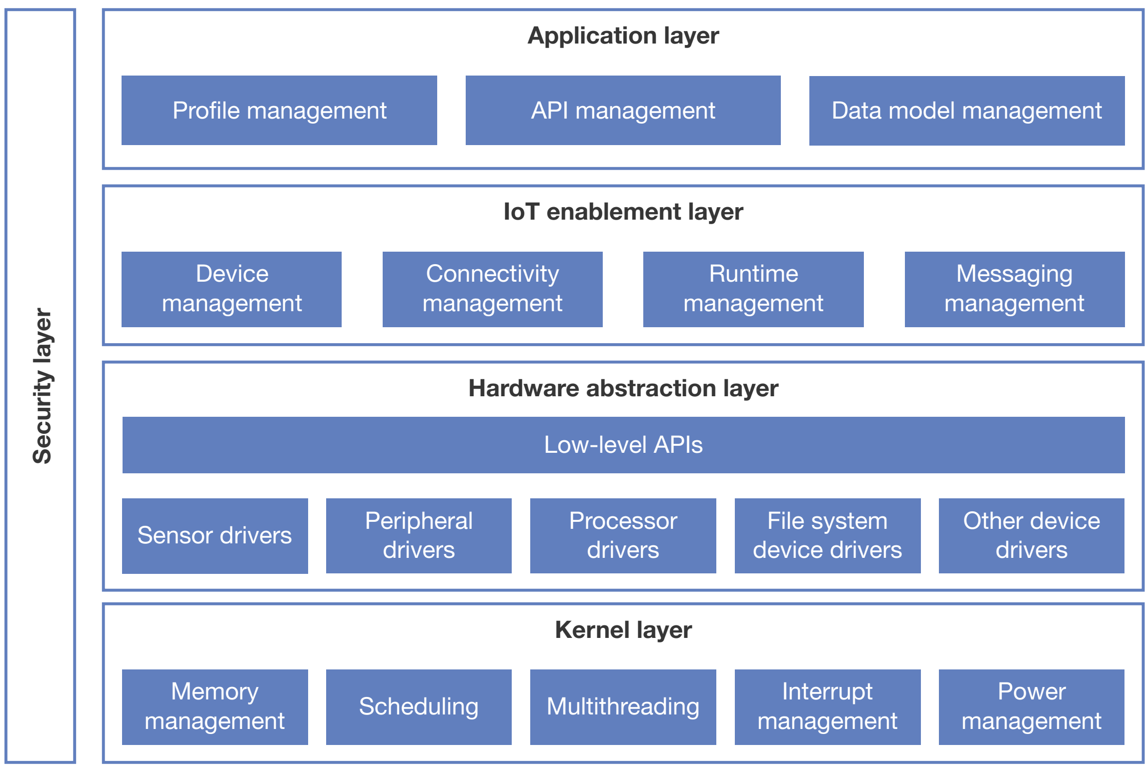
› **The IoT enablement layer enables foundational capabilities for IoT scenarios.** This layer has four components. Device management supports device resource discovery, status check, and life-cycle management. Connectivity management supports long-range protocols such as LTE-MTC and NB-IoT and short-range protocols like BLE, NFC, LoRa, Zigbee, Wi-Fi, 6LowPan, Thread, IPv4/IPv6, and industrial protocols. Runtime management supports JavaScript runtime libraries, C/C++ runtime libraries, and other system runtimes. Messaging management supports diverse application transfer protocols, such as XMPP, CoAP, MQTT, HTTP, and LWM2M.[10]

› **The application layer supports IoT application development.** This layer has three components. Profile management streamlines device configurations through prebuilt and customizable hardware profiles for IoT applications in different horizontal contexts or vertical segments, such as smart home, manufacturing, and utilities. API management manages high-level APIs for developers to access hardware resources. Data model management provides a unified approach to defining data models of both logical objects and physical device properties, such as temperature in the thermometer.

FORRESTER®
© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

› **The security layer protects IoT devices from end to end.** Given the pervasiveness of IoT devices beyond company boundaries, a Zero Trust security model is critical to an IoT device OS to ensure end-to-end security.[11] Identification and authentication validates user access. Discretionary access controls restrict access to data objects in the file system. Management policy enforcement on resources such as memory and CPU protects devices from running out of resources under DDoS attacks. Data encryption and residual information protection secure data storage and access.[12] Support for protocols like TLS and DTLS provides privacy and data integrity in networks.[13]

FIGURE 2 Reference Architecture Of The Internet Of Things Device Operating System Capability Model



MAP THE DNA OF IOT DEVICE OSES TO APPLICATION SCENARIOS TO ADDRESS BUSINESS NEEDS

Regardless of whether an IoT device OS is open source or commercial, two dimensions determine how feasible each OS is for your business requirements: the nature of the OS and the real-time needs of application scenarios (see Figure 3). Analysis of these two dimensions shows that:

› **Not all IoT device OSes are created equal.** The functionality and market position of various offerings determine their DNA, which falls into three categories: OS foundation, integrated OS, and OS framework. IoT device OS foundations such as Raspbian and NuttX focus on kernel

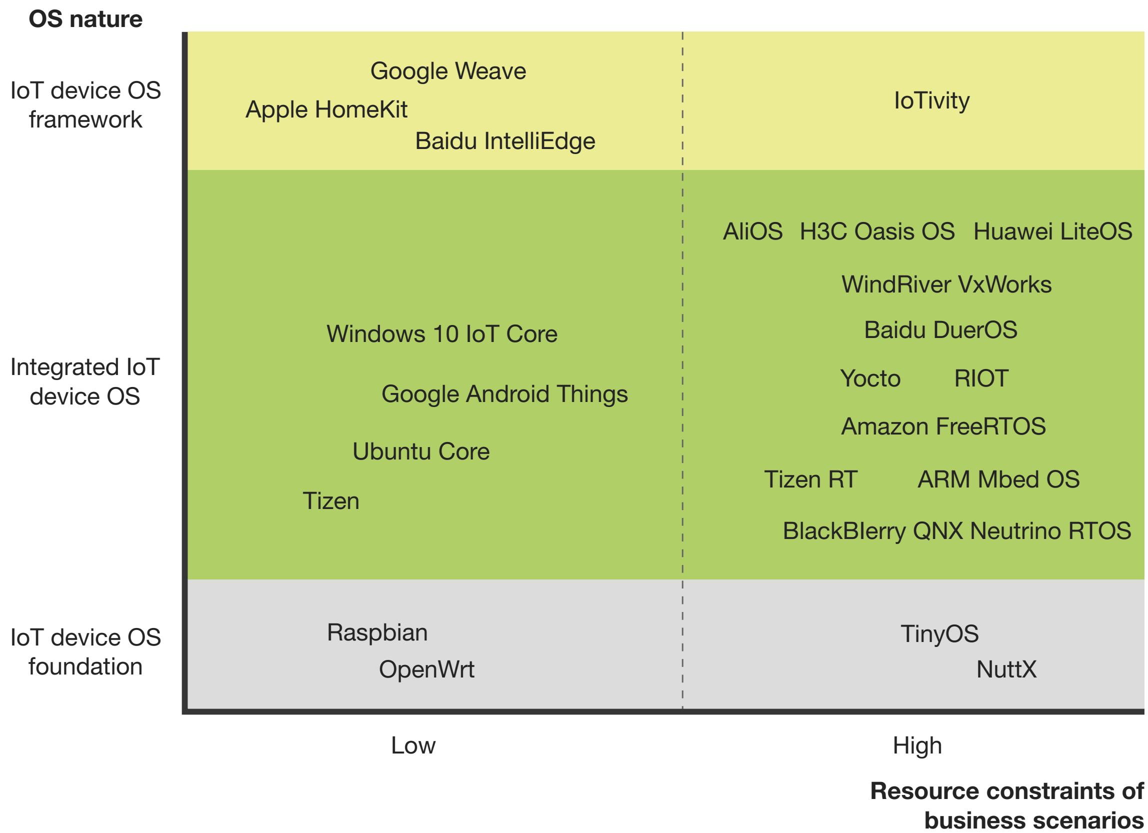
© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

and hardware abstraction. Integrated IoT device OSes, such as Windows 10 IoT Core, Amazon FreeRTOS, AliOS Things, and Huawei LiteOS, support all architecture layers with different capabilities. IoT device OS frameworks focus on the IoT enablement and application layers.

- **Resource constraints is the most critical factor for your business case.** Many business scenarios for IoT device OSes are severely resource-constrained; onboard batteries or solar panels can only supply limited amounts of power. And typical sensor devices, such as those in manufacturing product lines or logistics networks, are equipped with 8-bit microcontrollers. The small physical size and low cost of devices limit system complexity, storage, and memory. In contrast, smart home devices or systems on edge networks that focus more on data processing and networking can be less resource-constrained.
- **The alignment of OS type with resource needs is key.** DevOps' starting point depends on the nature of the IoT device. For manufacturers of consumer electronics that already have an OS with limited IoT capabilities embedded in appliances, IoT device OS frameworks like Google Weave and IoTivity can be a better choice. Key technical features to look at depend on the resource needs of business scenarios. RTOS-based IoT device OSes like ARM Mbed OS and WindRiver VxWorks are more suitable for highly resource-constrained scenarios.

FORRESTER®
© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

FIGURE 3 Mapping The Nature Of The Internet Of Things Device Operating System To Application Scenarios



Note: The position of a vendor offering within a specific segment is for illustrative purposes only and is not intended as a comparison.

VIEW THE OFFERINGS HOLISTICALLY TO MAKE THE RIGHT DECISION

EA pros should realize the architectural complexity of creating IoT device OS software from scratch. To accelerate digital transformation, they should consider commercial solutions and mainstream open source frameworks by aligning technology decisions with their business scenarios. Based on the nature and resource requirements of various IoT device OS offerings, Forrester defines three market segments: IoT device OS framework, full-stack IoT device OS, and embedded device OS. The full-stack IoT device OS and embedded device OS segments both include integrated IoT device OS and IoT device OS foundation functionality.

› **IoT device OS framework.** These products only offer an IoT framework for device-side operations and management on the enablement and application layers with related security features. They do not provide underlying OS capabilities on hardware abstraction and kernel layers and will need a

© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

base OS for hardware support. Frameworks like IoTivity from the Open Connectivity Foundation (OCF) and Baidu IntelliEdge cover broad IoT application scenarios; Apple HomeKit and Google Weave mainly address smart home requirements (see Figure 4).[14]

- **Full-stack IoT device OS.** These products mainly target nonembedded or less resource-constrained scenarios. They cover a variety of OS architectures and functions. Microsoft uses Windows 10 for its IoT Core; it provides built-in support and integration options for a wide range of protocols, enabling more than 5,000 enterprises' IoT initiatives. These include Beckhoff's industrial and manufacturing automation systems and Philips Healthcare's connected ultrasound machines. Canonical architects its offering on Ubuntu; more than 500 customers use Ubuntu products, such as Fingbox for home networking security devices and Intersection for showcasing media information, to facilitate digital businesses (see Figure 5).
- **Embedded device OS.** These products mainly target embedded working environments, which normally have strict constraints on hardware resources and high demand for real-time capabilities. These offerings effectively enable firms worldwide in their digital business. In the smart city market, Huawei's LiteOS powers Quectel's NB-IoT communication modules in Middle East and the Chinese city of Yingtan; H3C's Oasis OS is enabling Qingdao Yilian's parking detectors in China. Among public cloud service providers, Amazon FreeRTOS helped NASA JPL with its robotic swarm prototype; AliOS from Alibaba Cloud provides interconnectivity for Vanke's home appliances (see Figure 6).

FIGURE 4 Representative IoT Device OS Offerings: IoT Device Framework

Vendor or organization	Offering name	Application coverage	Open source	Memory size	Protocol support: application data transfer	Protocol support: transport connectivity
Apple	HomeKit	Specialized (smart home)	No		HomeKit Access Protocol	Wi-Fi, Bluetooth Low Energy (BLE), Bluetooth Smart, HTTP, ethernet
Baidu	IntelliEdge	General-purpose	Yes	RAM: 20 MB	MQTT	N/A
Google	Weave OpenWeave	Specialized (smart home)	Yes	N/A	CoAP	Wi-Fi, Thread
Linux Foundation, Open Connectivity Foundation	IoTivity	General-purpose	Yes	RAM: 600 KB	CoAP	BLE, Wi-Fi, Z-Wave

© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

FIGURE 5 Representative IoT Device OS Offerings: Full-Stack IoT Device OS

Vendor or organization	Offering name	Application coverage	Open source	Memory size	Protocol support: application data transfer	Protocol support: transport connectivity
Canonical	Ubuntu Core	General-purpose	Yes	RAM: 128 MB Image: 350 MB	CoAP, MQTT, AMQP	Modbus, CAN, Zigbee*
Google	Android Things (formerly Brillo)	General-purpose	Yes	RAM: 512 MB	CoAP	Wi-Fi, BLE, LoWPAN (incl. Thread)
Microsoft	Windows 10 IoT Core	General-purpose	Yes	RAM: 512 MB Storage: 2 GB	MQTT, AMQP, HTTPS	Cellular (3G, 4G, 4G LTE, LTE-A, 5G), Bluetooth, Wi-Fi, NFC, IPv4, IPv6
	Windows 10 IoT Enterprise	General-purpose	Yes	RAM: 1 GB to 2 GB Storage: 32 GB	MQTT, AMQP, HTTPS	Cellular (3G, 4G, 4G LTE, LTE-A, 5G), Bluetooth, Wi-Fi, NFC, IPv4, IPv6
	Azure Sphere	General-purpose	Yes	RAM: 4 MB ROM: 16 MB	MQTT, AMQP, HTTPS	Wi-Fi, IPv4
Raspberry Pi Foundation	Raspbian	Specialized (Raspberry Pi)	Yes	Image: 350 MB to 4 GB	MQTT, CoAP	BLE, Wi-Fi, 5G
Tizen Association	Tizen	Specialized (connected cars, consumer electronics)	Yes	ROM: 2 GB	CoAP	IPV4, IPV6, Zigbee, Thread, Wi-Fi, 6LoWPAN, BLE

*All protocols available to a Linux system apply, installable through software repositories and third-party downloads.

© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

FIGURE 6 Representative IoT Device OS Offerings: Embedded IoT Device OS

Vendor or organization	Offering name	Application coverage	Open source	Memory size	Protocol support: application data transfer	Protocol support: transport connectivity
Alibaba Cloud	AliOS	General-purpose	Yes	RAM: 64 KB ROM: 1 MB	MQTT, CoAP, XMPP	BLE, NFC, LoRa, Zigbee, Wi-Fi, IPv4, IPv6, 6LowPAN
Amazon Web Services	Amazon FreeRTOS	General-purpose	Yes	RAM: 16 KB or more ROM: 10 KB*	MQTT	Wi-Fi, ethernet
Apache Mynewt	Mynewt OS	General-purpose	Yes	N/A	CoAP	BLE, Wi-Fi, LoRaWAN Wi-Fi
ARM	Mbed OS	General-purpose	Yes	RAM: ~10 KB ROM: ~6 KB	MQTT, CoAP	BLE, ethernet, Wi-Fi, LoRa, NFC, 6LoWPAN (including Thread), cellular (NB-IoT, 4G LTE, CAT-M1, GPRS, 3G WCDMA)
Baidu	DuerOS	Specialized (connected cars and smart home)	Yes	N/A	MQTT	N/A
BlackBerry	QNX Neutrino RTOS	Specialized (connected cars)	No	RAM: 32 KB	N/A	N/A
Contiki		General-purpose	Yes	RAM: 2 KB ROM: 60 KB	CoAP	Ethernet, 6LoPWAN

*Amazon FreeRTOS is optimized for microcontrollers faster than 25 MHz and with more than 64 KB RAM, assuming all available libraries, including TLS, are running on the application microcontroller. If the communication and crypto stack (except for MQTT) is offloaded onto the networking processor, the microcontroller only needs 10 MHz and 16 KB. These values are approximate, as factors like MCU architecture, compiler, and compiler optimization level may affect speed and RAM requirements. Amazon FreeRTOS needs 128 KB of program memory per executable image stored on the microcontroller. For OTA update functionality, two executable images must be stored in program memory at the same time.

FORRESTER®
© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

FIGURE 6 Representative IoT Device OS Offerings: Embedded IoT Device OS (Cont.)

Vendor or organization	Offering name	Application coverage	Open source	Memory size	Protocol support: application data transfer	Protocol support: transport connectivity
H3C	Oasis OS	General-purpose	Yes	RAM: 5 KB to 256 MB ROM: 10 KB to 256 MB	CoAP, MQTT	BLE, LoRa, LwIP, cellular (NB-IoT)
Huawei	LiteOS	General-purpose	Yes	RAM: 6 KB ROM: 14 KB	CoAP, MQTT	IPv4, IPv6, BLE, Wi-Fi, cellular (GPRS, UMTS, eMTC, NB-IoT, LTE), 6LoWPAN
Linux Foundation	Zephyr	General-purpose	Yes	RAM: 8 KB	MQTT, CoAP	Bluetooth, BLE, Wi-Fi, IPv4, IPv6, 6LoWPAN
NuttX project	NuttX	General-purpose	Yes	RAM: 5 KB to 20 KB	N/A	Ethernet, 6LoWPAN
OpenWrt project	OpenWrt	Specialized (routers and smart home)	Yes	RAM: 0.9 KB ROM: 1.9 KB	CoAP, MQTT	Wi-Fi
RIOT project	RIOT	General-purpose	Yes	RAM: 1.5 KB ROM: 5 KB	CoAP	6LoWPAN, IPv6
TinyOS Alliance	TinyOS	General-purpose	Yes	RAM: 4 KB to 10 KB	CoAP, MQTT	IEEE 802.15.4 (wireless radio network by transceivers like CC2420)
Tizen Association	Tizen RT	Specialized (low-end consumer electronics)	Yes	RAM: 2 MB ROM: 16 MB	CoAP	IPv4, IPv6, Thread, Wi-Fi, 6LoWPAN, BLE
Wind River	VxWorks	General-purpose	Yes	RAM: 20 KB	CoAP, MQTT	Bluetooth, SocketCAN
Yocto project	Yocto	General-purpose	Yes	N/A	MQTT, AMQP	N/A

FORRESTER®
© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

Recommendations

Take A Data-Driven Approach To Embrace IoT Device OS

The landscape of IoT device OS software is fragmented, with a wide variety of commercial and open source offerings. These offerings cover different capabilities in the five architectural layers; most are taking an open source strategy to facilitate ecosystem evolution, making this market even more dynamic. To navigate the tricky waters of IoT device OSes, address business and technical challenges, and help their companies embrace the right IoT device OSes, EA pros must take a data-driven approach. You must:

› **Identify the data bottleneck for your business pain points.** IoT integration scenarios are complex, and there is no one-size-fits-all solution. EA pros must collaborate with business stakeholders and technology leaders to pinpoint the bottlenecks from a data perspective. Does your chief operations officer need specific operational data that is hidden in the devices of the supply chain? Is your infrastructure and operations director complaining that the aggregation of sensor data is not as real-time as line-of-business leaders expect? Answer these questions to make your business case of IoT device OS much more complete.

› **Define the data model for device-side interoperability.** Data from connected devices comes in a wide array of formats and models, and the interoperability of IoT systems is more complicated than you might expect.[15] The device data model is key to consistent data exchange and reusable integration patterns, and EA pros should keep a close watch on key initiatives of relevant organizations. OCF created oneIoTa, an open online tool to encourage the design of interoperable IoT device data models.[16] OMA SpecWorks offers smart object guidelines that provide an object model for interoperability between devices using the CoAP protocol.

› **Leverage the data synergy of cloud-based IoT software platforms.** Digital giants like Alibaba Cloud, Amazon Web Services, Baidu, Google, Huawei, and Microsoft are all making strategic investments in both IoT device OS software and IoT services over their public cloud platforms, which implies better data interoperability on the transport and application layers. You can also use these vendors' other platform services, such as machine learning, computer vision, and even blockchain, to maximize the business value realization from your IoT devices.

© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

Engage With An Analyst

Gain greater confidence in your decisions by working with Forrester thought leaders to apply our research to your specific business and technology initiatives.

Analyst Inquiry

To help you put research into practice, connect with an analyst to discuss your questions in a 30-minute phone session — or opt for a response via email.

Learn more.

Analyst Advisory

Translate research into action by working with an analyst on a specific engagement in the form of custom strategy sessions, workshops, or speeches.

Learn more.

Webinar

Join our online sessions on the latest research affecting your business. Each call includes analyst Q&A and slides and is available on-demand.

Learn more.

Forrester's research apps for iOS and Android.
Stay ahead of your competition no matter where you are.

Supplemental Material

SURVEY METHODOLOGY

The Forrester Analytics Global Business Technographics® Data And Analytics Survey, 2018, was fielded in February and March 2018. This online survey included 2,879 respondents in Australia, Canada, China, France, Germany, India, the UK, and the US from companies with 100 or more employees.

Forrester Analytics Business Technographics ensures that the final survey population contains only those with significant involvement in the planning, funding, and purchasing of business and technology products and services. Research Now fielded this survey on behalf of Forrester. Survey respondent incentives include points redeemable for gift certificates.

Please note that the brand questions included in this survey should not be used to measure market share. The purpose of Forrester Analytics Business Technographics brand questions is to show usage of a brand by a specific target audience at one point in time.

FORRESTER®
© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

DATA ACCESS AND SERVICES

You can perform custom analyses and profiling with the Forrester Analytics Global Business Technographics Data And Analytics Survey, 2018. Parameters available for analysis may include:

- Demographics: such as job function and level of seniority.
- Firmographics: such as industry, company size, and revenue growth.
- Technology behavior: such as general priorities.
- Special segments: such as unique groups defined by combining parameters.

For data subscribers, please contact your data insights director or manager with any questions regarding the use of the Forrester Analytics Global Business Technographics Data And Analytics Survey, 2018, or custom data analyses. If you do not have a direct point of contact, please pose all inquiries to the Technographics Team, who can be reached at datarequest@forrester.com.

For non-data subscribers, please contact the Technographics Business Development Team with any questions regarding the use of this data or custom data projects at data@forrester.com.

COMPANIES INTERVIEWED FOR THIS REPORT

We would like to thank the individuals from the following companies who generously gave their time during the research for this report.

Alibaba Cloud

Amazon Web Services

Canonical

H3C

Huawei

Microsoft

Endnotes

[1] For example, real-world data from IoT systems can delight customers with usage insights that drive more contextual digital experiences. See the Forrester report "Use IoT To Link Physical Operations And Digital Business."

[2] Connectivity options include 3G, 4G, and emerging 5G mobile networks as well as improved wireless technologies like Bluetooth Low Energy (BLE), ZigBee, Long Range (LoRa), and NB-IoT. IoT digitizes and automates manual activities in the real world. See the Forrester report "Untangle Your IoT Strategies."

[3] IoT software stack architecture contains five layers; IoT device OSes span four of them: the connectivity, security, management, and compute and analytics layers. See the Forrester report "Boost Digital Business With The Internet Of Things."

[4] Some IoT devices connect directly to cloud servers, while others connect to nearby gateway devices or use peer-to-peer mesh networking to access the internet. See the Forrester report "Design Your Integration Architecture For The Internet Of Things."

[5] Security teams are dealing with an increasingly sophisticated security threat landscape. See the Forrester report "The State Of IoT Security 2018."

© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

[6] Connected sensor and electromechanical devices carry a far different set of requirements than traditional technology infrastructure. See the Forrester report "Implementation Timelines Will Drive Your IoT Low-Power Wireless WAN Technology Decision."

[7] Running multiple threads will also require more power. Single-threaded multicore implementations can be a better choice if the ratio of performance to power efficiency is crucial.

[8] FreeRTOS uses idle task hook to put the microcontroller in a low-power state. Source: FreeRTOS (https://www.freertos.org/low-power-tickless-rtos.html).

[9] A universal asynchronous receiver/transmitter (UART) is the microchip with programming that controls a computer's interface to its attached serial devices. Source: Margaret Rouse, "UART (Universal Asynchronous Receiver/Transmitter)," TechTarget, February 2011 (https://whatis.techtarget.com/definition/UART-Universal-Asynchronous-Receiver-Transmitter).

Serial peripheral interface (SPI) is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. Source: Ardruino (https://www.arduino.cc/en/Reference/SPI).

The I2C bus was designed by Philips in the early 1980s to allow easy communication between components on the same circuit board. Source: I2C-Bus (https://www.i2c-bus.org/).

[10] IoT devices and applications require diversified protocols on different layers of the Open Systems Interconnection model to support various interconnectivity options. See the Forrester report "Boost Digital Business With The Internet Of Things."

[11] Zero Trust enables companies to manage the risk by, for example, creating microperimeters around IoT devices. See the Forrester report "Zero Trust Security: A CIO's Guide To Defending Their Business From Cyberattacks."

[12] Residual information protection is a security feature that insures that whenever a resource is allocated or freed, the content of the resource can be made unavailable to others' processes. Source: Robert Day and Michael Slonosky, "Securing connected embedded devices using built-in RTOS security," Military Embedded Systems, March 9, 2015 (http://mil-embedded.com/articles/securing-connected-embedded-devices-using-built-in-rtos-security/).

[13] Transport layer security (TLS) uses TCP and datagram TLS (DTLS) uses UDP. Source: Stack Overflow (https://stackoverflow.com/questions/15331294/difference-between-dtls-and-tls).

[14] The Open Connectivity Foundation (OCF) is making the IoT attainable by establishing the necessary interoperability standard for connected devices, enabling them to discover and communicate with one another regardless of manufacturer, OS, chipset, or physical transport. Source: Open Connectivity Foundation (https://openconnectivity.org/foundation/faq).

[15] The structure of the IoT means that there are different levels of interoperability. See the Forrester report "Brief: Bringing Interoperability To The Internet Of Things."

[16] The web-based oneIoTa tool enables users to create simple models for IoT devices using RAML and JSON. Source: Open Connectivity Foundation (https://openconnectivity.org/developer/oneiota-data-model-tool).

FORRESTER®
© 2019 Forrester Research, Inc. Unauthorized copying or distributing is a violation of copyright law.
Citations@forrester.com or +1 866-367-7378

FORRESTER®
CHALLENGE THINKING. LEAD CHANGE.

We work with business and technology leaders to develop customer-obsessed strategies that drive growth.

PRODUCTS AND SERVICES

- Core research and tools
- Data and analytics
- Peer collaboration
- Analyst engagement
- Consulting
- Events

Forrester's research and insights are tailored to your role and critical business initiatives.

ROLES WE SERVE

Marketing & Strategy Professionals
CMO
B2B Marketing
B2C Marketing
Customer Experience
Customer Insights
eBusiness & Channel Strategy

Technology Management Professionals
CIO
Application Development & Delivery
› Enterprise Architecture
Infrastructure & Operations
Security & Risk
Sourcing & Vendor Management

Technology Industry Professionals
Analyst Relations

CLIENT SUPPORT

For information on hard-copy or electronic reprints, please contact Client Support at +1 866-367-7378, +1 617-613-5730, or clientsupport@forrester.com. We offer quantity discounts and special pricing for academic and nonprofit institutions.

Forrester Research (Nasdaq: FORR) is one of the most influential research and advisory firms in the world. We work with business and technology leaders to develop customer-obsessed strategies that drive growth. Through proprietary research, data, custom consulting, exclusive executive peer groups, and events, the Forrester experience is about a singular and powerful purpose: to challenge the thinking of our clients to help them lead change in their organizations. For more information, visit forrester.com.