The Past, Present, and Future of the **Internet of Things**

By Stuart Cording (Elektor)

The Internet of Things (IoT), like many technologies before it, resulted in a hype similar to adding web services to your business during the dot-com boom. However, by giving this network of massively connected microprocessor-based devices a name, it afforded focus that resulted in protocols and cloud services dedicated to the needs of this emerging tech. Security and upgradeability remain weaknesses, but there are solutions on the horizon.

2010 *Oracle Corp. acquires Sun Microsystems*

2010 *Large Hadron Collider creates first mini Big Bang*

2011 *The passing of Steve Jobs*

Our industry didn't really need a name for embedded applications that are connected to the Internet. However, when Kevin Ashton coined the term "Internet of Things" (IoT) back in 1999 [1], the designation helped focus minds around the arising challenges. Engineers had connected microcontrollers (MCUs) to the Internet for as long as there had been modems and Ethernet chipsets. One of the first IoT demonstrators was built by John Romkey for the 1990 Interop show in San Jose, California [2]. Combining a laptop with a retro toaster, this humdrum home appliance became Internet-capable.

The Past

The spark that resulted in Kevin Aston's terminology was the realization that people were providing the large majority of the data available on the Internet. His research into radio-frequency identification (RFID) tags resulted in the hypothesis that computers would automatically track items and collect data. This would reduce the risk of data entry mistakes caused by humans. In fact, his preferred terminology is "Internet *for* Things [3]." This is not a language subtlety but rather an acceptance that some *things* will not need the full performance of, or be able to power long term, the Ethernet and Wi-Fi interfaces commonly used in laptops, PCs, and smartphones. Instead, interfaces and transport protocols would be needed that are better suited to systems requiring a long battery life and transferring quantities of data measured in bytes rather than megabytes.

The concept of machines communicating with one another was also not new. Machine-to-machine (M2M) communication was already a term, but it stemmed from the use of wired or cellular telephone networks to enable point-to-point connections. Siemens had launched their *M1* GSM module for precisely this purpose back in 1996 [4]. Then there were terms such as *pervasive computing*, *ubiquitous computing (UbiComp)*, or even *sensor networks*. Somehow, the term IoT seemed specific enough without being too generic for the broader community, such as financial

Looking ahead to the future of IoT, security and upgradability will need to be addressed.

investors who wanted to jump on the next tech bandwagon. As with any technology, financing is critical. Without investment and the subsequent build-up of resources, even the best technologies will die.

MCUs were obviously part of the puzzle. The 2000s had seen a growth in new products (see "The Microcontroller Boom") and suppliers, with some specializing in low-power. The next puzzle piece was connectivity. Clearly, IoT products would need to be gathering sensor data and not always in a location where new power and data cables could be laid or routed. Radio technology, such as Bluetooth, had been made possible due to the availability of the license-free industrial, scientific, and medical (ISM) band. Established in the United States in the 1940s, the availability of the 2.45 GHz ISM band opened the way for microwave ovens and diathermy-based medical equipment. CMOS-based radio transceivers enabled Bluetooth basebands to be manufactured on the same, low-cost fabrication processes as other semiconductors, making them economically viable.

The Present

Technology using the ISM's 433 MHz (Europe, Middle East, Africa) and 915 MHz (US) bands also emerged, such as LoRa and Zigbee. Others also exist, making use of the other license-free bands available around the world (**Figure 1**). For those looking to use existing infrastructure, the cellular network is another choice. With 5G in the deployment phase, the architects have accommodated the needs of mass IoT connectivity by supporting more than 100,000 nodes per square kilometer, up from 2,000 under 4G. Recognizing the need for extended battery life, developers can use Narrowband IoT (NB-IoT, also known as LTE Cat NB2) chipsets and modules that are matched to the low-power, low data rate needs of IoT applications.

Figure 1: Consumers have an array of smart home IoT devices to choose from, using a mixture of license-free ISM band radio technologies.

Figure 2: Farmers on the island of Jersey make use of a LoRa radio network for weather and environmental sensing. (Source: Shutterstock)

2012 *Google acquires Motorola Mobility*

2012 *Ultra High Definition standard*

2012 *Raspbian free OS released*

Jersey, one of the channel islands near the northwest coast of France, is a prime example of where IoT is happening at scale. Thanks to its small size and independence in decision-making, it is a great testing ground for IoT deployments. Known for farming Jersey Royal potatoes, it has used LoRaWAN technology for its weather stations, improving on the previous radio-based system used (**Figure 2**). The collected data is uploaded to a disease forecasting platform, allowing farmers to spray more judiciously to combat potato blight [5]. Eight gateways cover 75% of the island, allowing additional LoRa sensors to monitor field conditions, optimizing irrigation, and reducing the amount of fertilizer applied to the fields. Tracking of farming vehicles, such as tractors and 4x4s, is also being discussed. This will likely use NB-IoT with the data used to help reduce fuel usage and improve driver behavior.

Another core piece of the puzzle is software. This requires those with an intimate knowledge of embedded systems, with their limited performance and memory capacity, and need to extend battery life, to work alongside cloud solutions providers, where there are, essentially, no such limits. One example is MQTT that emerged from a requirement for a bandwidth-constrained communication protocol to monitor sensors on oil pipelines [6]. Unlike HTTP, used to serve web pages to computers and smartphones, MQTT is event-driven, allowing IoT devices to share data as and when it becomes available. Client sensing devices post their data to a broker, running on a server. Other devices, such as backend servers or mobile devices, can register as subscribers, receiving only the data they are interested in. Built upon TCP/IP, it can also use secure protocols such as SSL/TLS to protect the data [7].

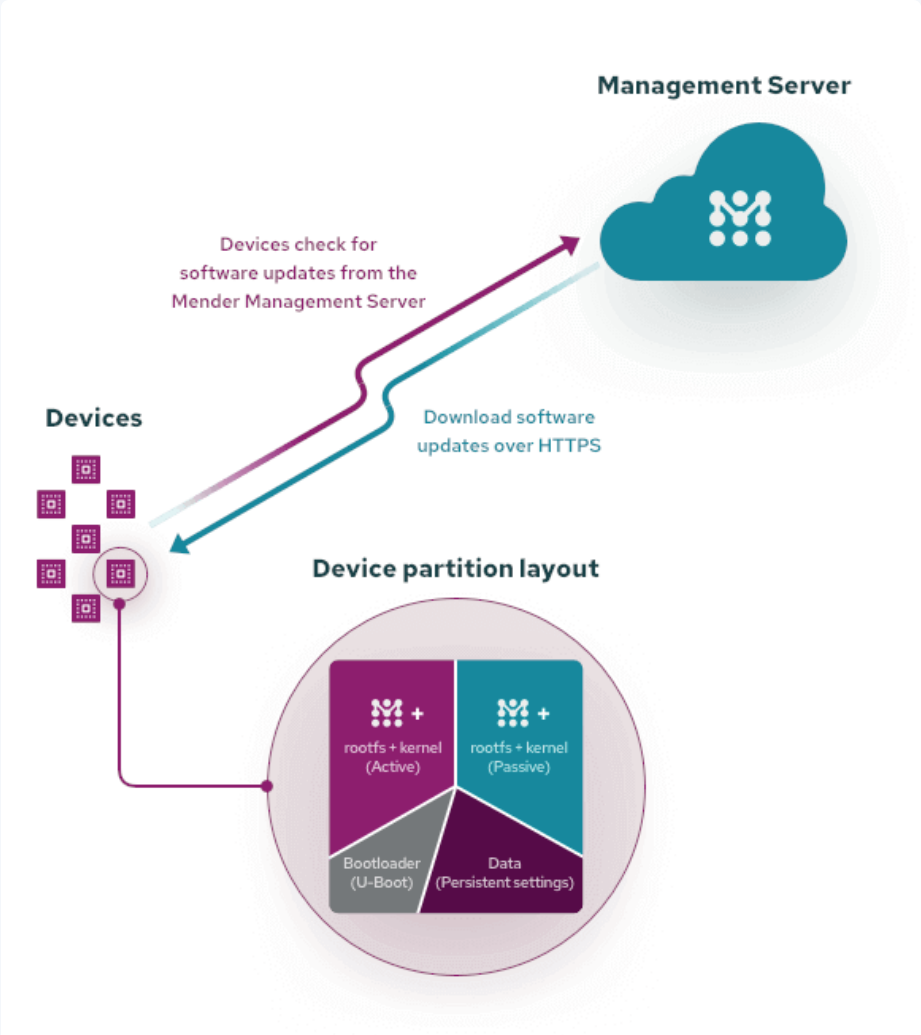

Figure 3: The Mender approach is useful for Linux-based IoT solutions. Using a dual rootfs partition avoids bricking the device when updating it. (Source: Mender)

Security and Upgradeability

Reusability is an essential part of the success of much of the IoT. Still, standardization drops significantly the further developers move away from the server to the infrastructure and sensors. This causes a range of challenges, from deploying updates to implementing security. Talking to Eystein Stenberg of Mender, his IoT business has benefitted from the growth in deployment of the Linux-based Raspberry Pi Zeros in industrial, infrastructure, and automotive applications. Until recently, keeping such devices up-to-date had been an enormous challenge, with each company building a roll-your-own solution.

Mender provides a platform that can roll out updates to Linux-based embedded devices, either to upgrade the operating system or simply patch smaller parts of a running system [8]. A dual-partition approach ensures that OS, kernel or library updates cannot inadvertently brick the system (**Figure 3**). The broad use of Yocto and Debian in such applications means that the process can be easily streamlined and deployed. However, rolling out updates to non-Linux processors, such as MCUs running on real-time kernels such as FreeRTOS, µC/OS, or embOS, remains a challenge.

Traditionally, update processes for MCUs, if offered at all, are a risky business that could result in the device becoming bricked (i.e., incapable of restarting due to a failed update). The IoT company Toit resolves this by implementing a virtual machine upon which your apps can be executed (**Figure 4**). Much

like running an app in VirtualBox or Parallels, a faulty app cannot crash other apps or the underlying virtual machine as they lie in isolated containers. Thus, should a single app update fail, the remainder of the code executing on the MCU continues to function unhindered. Currently, support is provided for the Espressif ESP32. The programming language for apps also has to be learned. It is similar to Python but, compared to MicroPython, is 20 times faster [9].

Companies like Mender and Toit have appeared because of IoT security or, more accurately, the lack of it. Security is an exceptionally complex field and, without suitable expertise, it is easy to become convinced that a convoluted system is also secure. Even with proper security, updates must be rolled out promptly after a vulnerability has been found. Looking ahead to the future of IoT, security and upgradability will need to be addressed. Semiconductor vendors, such as NXP, with their EdgeLock [10] technology, and the OPTIGA [11] solutions from Infineon, can protect the connections of IoT devices to the cloud. However, these are of little use if hackers find vulnerabilities in the chip-to-chip communication on the circuit board, allowing them to inject their own data or capture cryptographic keys.

The Future

So, what does the future hold for IoT? Mender's Stenberg thinks that Linux-capable SoCs could drop so low in price as to make them attractive even for the low-cost IoT nodes. That could potentially make MCU-based devices using proprietary software and real-time kernels a thing of the past. With it, standardization in software and software updates could also come about, reducing the associated costs for updating and maintaining networks of devices. But, who really knows. What is certain is that the future is hard to predict with the level of innovation bursting out of our electronics industry. Perhaps we'll know and share more when we celebrate 70 years of *Elektor*!

210544-01

About the Author

Stuart Cording is an engineer and journalist. With more than 25 years of experience in the electronics industry, you can read many of his Elektor articles at www.elektormagazine.com/cording.

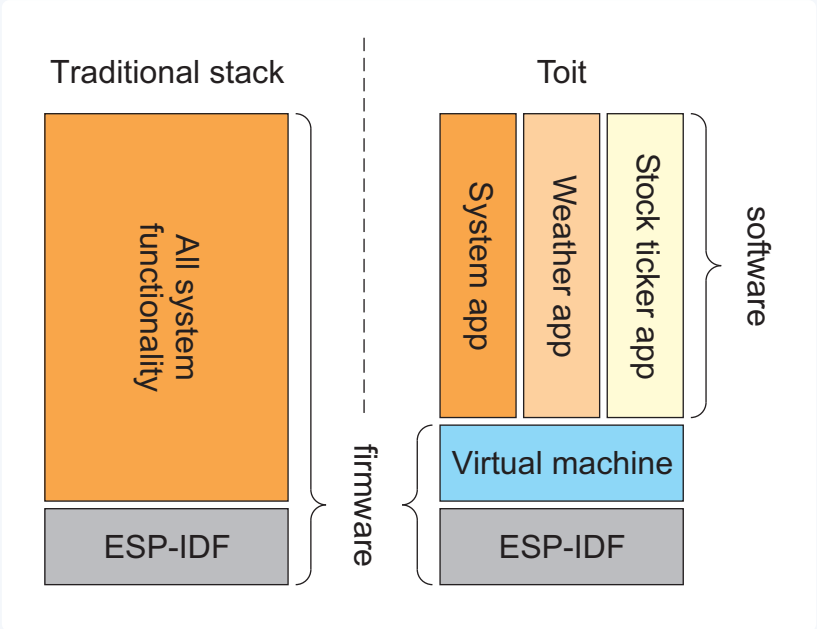

Figure 4: For non-Linux-based IoT, Toit's virtualization on the ESP32 stores each app in its own container.

WEB LINKS

[1] K. L. Lueth, "Why the Internet of Things is called Internet of Things: Definition, history, disambiguation," IoT Analytics, December 2014: https://bit.ly/3itumdn
[2] J. Elder, "The internet's first thing – John Romkey's 'smart' toaster," Avast, September 2019: https://bit.ly/39Z0LUy
[3] G. Press, "A Very Short History Of The Internet Of Things," Forbes, June 2014: https://bit.ly/3FdcU6G
[4] "GSM-Modul M1," ComputerWoche, March 1996: https://bit.ly/3DdjHvn
[5] "The Jersey Royal Company," Digital Jersey: https://bit.ly/3l3IDPw
[6] Transcript of IBM Podcast on MQTT: https://ibm.co/3A6wcqy
[7] M. Yuan, "Getting to know MQTT," IBM, May 2017: https://ibm.co/3ovhZS2
[8] "How it works," Mender: https://bit.ly/2WFd45m
[9] "Frequently asked questions," Toit: https://bit.ly/3mlAzsT
[10] "Secure Authentication and Anti-Counterfeit Solutions," NXP: https://bit.ly/3D9tqTu
[11] "OPTIGA™ embedded security solutions," Infineon: https://bit.ly/3FaBpBz