Internet of Things? Hold my beer, I got this: ARM crafts OS to rule them all

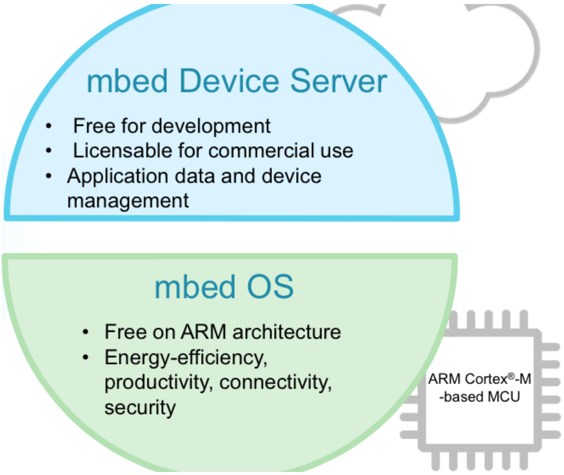
New mbed operating system tries to pave over chip rivalries to lure in IoT startups

1 Oct 2014 at 17:15, Chris Williams

20

ARM TechCon ARM will today announce a new operating system called mbed OS: it seeks to smooth over all the differences between various competing system-on-chips so that high-level applications can talk to sensors and other gizmos whether they're using silicon from Atmel, Marvell, ST, Freescale, NXP, and so on.

The software will be revealed at ARM TechCon 2014 in Santa Clara, California, on Wednesday morning, although it's not yet fully available. But here's what we've been told about it.


mbed Device Server
• Free for development
• Licensable for commercial use
• Application data and device management
mbed OS
• Free on ARM architecture
• Energy-efficiency, productivity, connectivity, security
ARM Cortex®-M -based MCU

It essentially comes in two parts: in the bottom half, there's the mbed OS kernel running on system-on-chips in little devices: this OS provides the drivers, reacts to stuff happening around the hardware, and communicates with the outside world. Programmers can write software to run directly on top of the kernel; this code controls the operation of the device.

Then in the top half of the stack, there's server-side software called the mbed Device Server that runs on much larger x86 or ARM-powered computers. This connects high-level applications, such as a website back-end, to the individual devices; it allows large bits of software to manage and draw from lots of little gadgets that don't, individually, have much in the way of intelligence.

An internet-connected gateway, which could be a smartphone, tablet or a little box in a cupboard, talks to devices over short-range comms, such as Wi-Fi or Bluetooth, and then routes that information through to an mbed Device Server. That server could be in the cloud, or could be a local machine. A device could, if it's capable, skip past the gateway stage and go straight to the server. It's a flexible architecture.

A lot of this stack, particularly at the lower level, draws from ARM's mbed community that's been going since 2009: this is a big pile of resources and toolchains for building ARM microcontroller firmware using C and C++. We're told more than 70,000 programmers are signed up on the mbed.org site, and have been working on thousands of projects for 30 or more prototyping and development boards supported by the mbed code.

Now, it appears, ARM's tying everything together, and packaged it so that high-level app developers can prototype stuff for ARM-powered sensors'n'gadgets faster – since there's a common, well-defined base from which to start.

The plan is to use open standards, such as HTTP and MQTT on top of TLS and DTLS, to deliver data securely between devices and whatever software is masterminding them. The software supports Bluetooth Smart; 2G, 3G, LTE and CDMA phone networks; Google Thread; Wi-Fi; and 6LoWPAN. It's understood ARM is not using an open-source library for its TLS cryptography.

The stack also hides away things like device availability: for example, high-level server-side software can request data from a sensor that may be sleeping to save energy, and eventually the information is received when the device is woken up, all of which is transparent to the high-level app. This simplifies the top level of the stack, which helps people write and release code to users a bit faster.

This may be upsetting news for engineers who enjoy getting their fingers dirty programming microcontrollers using assembly language, but the point of the mbed project has been to abstract the chaotic world of SoC design away from developers – leaving a cleaner, common interface to work with.

"Compilers these days are quite good," ARM's CTO, and company cofounder, Mike Muller told us.

"It's no longer the 1980s where you have to use assembly language; you can do it more efficiently now, and development time really matters."

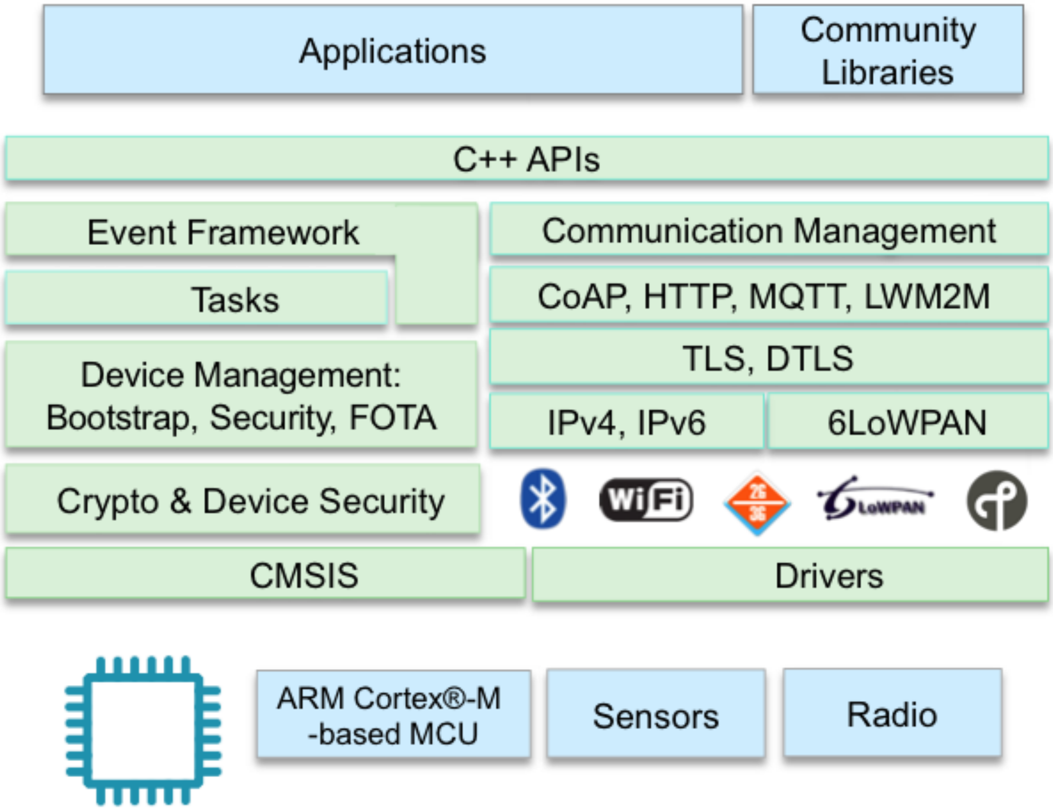
Under the hood

The new mbed OS is designed for the Cortex-M family, ARM's series of 32-bit microcontroller cores. These can cost about $5 to $10 each, and range from the tiny M0 to the beefier M4 and M7. The mbed OS is not a port of an existing kernel, such as the L4 microkernel found running on the iPhone 6's ARM-based crypto-coprocessor, QNX found in various car dashboards, nor the ubiquitous ARM GNU/Linux and uClinux.

We're told the new OS uses ARM's code, plus drivers and source contributed by hardware manufacturers and the mbed community. As you'd expect from a microcontroller OS, it's small enough to fit within the constraints of a Cortex-M SoC. A typical M0+ package will have between 8KB and 128KB of flash storage, and up to 16KB of RAM, while running at up to 48MHz; your common-or-garden M4 SoCs go up to 180MHz, and include 32KB to 2MB of flash and up to 256KB of SRAM. We're assured the mbed OS fits within the flash space of its target devices.

The OS is event driven rather than a classic real-time OS, Zach Shelby, ARM's director of technical marketing for the Internet of Things, told us. What that means is that it is not a complex preemptive kernel that divides the processor into time slices; there's no context switching or timers with deadlines. Instead, mbed OS sleeps until an interrupt from a sensor or other peripheral wakes it up, it handles the event, makes a decision whether to trigger a signal higher up the stack, and then falls to sleep again.

If you're used to microcontroller-level development, this will come as no surprise. It's just good sense in a resource-constrained environment; a design that's supposed to save power. "Batteries don't follow Moore's Law," Shelby said, reminding us that today's mobile processors are still constrained by their tiny power sources.

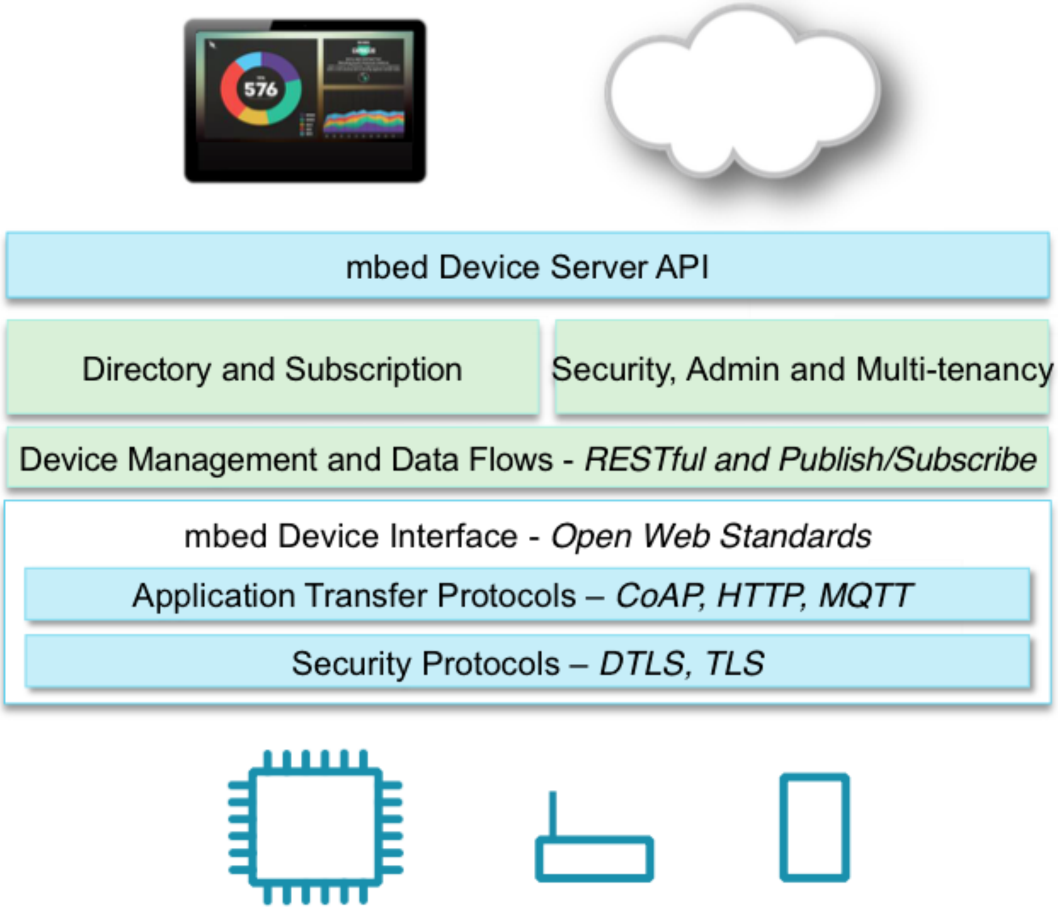


mbed OS ... the components of the operating system

Above is ARM's diagram of the layers in mbed OS; essentially, as a developer, you fill in the blue blocks at the top: the applications and libraries. Below is the makeup of the mbed Device Server, which abstracts away the lower levels for software that talks to iOS and Android apps, or other services.

The idea here is so that people who are handy with C++, JavaScript, HTML, Swift and other languages for phones, tablets and desktops can prototype and build applications for fiddly hardware ultimately hidden away under the mbed stack. These programmers don't need to know about the undocumented registers and control bits, and the rivalries, plaguing the system-on-chip world; it's abstracted away by mbed, which tries to pave over the fragmentation.

“Today’s Internet of Things largely exist in isolation, and it has been impossible to realize a truly interconnected world where devices are interoperable with many different cloud services,” said Krisztian Flautner, ARM's general manager of its IoT business.


576
mbed Device Server API
Directory and Subscription
Security, Admin and Multi-tenancy
Device Management and Data Flows - *RESTful and Publish/Subscribe*
mbed Device Interface - *Open Web Standards*
Application Transfer Protocols – *CoAP, HTTP, MQTT*
Security Protocols – *DTLS, TLS*

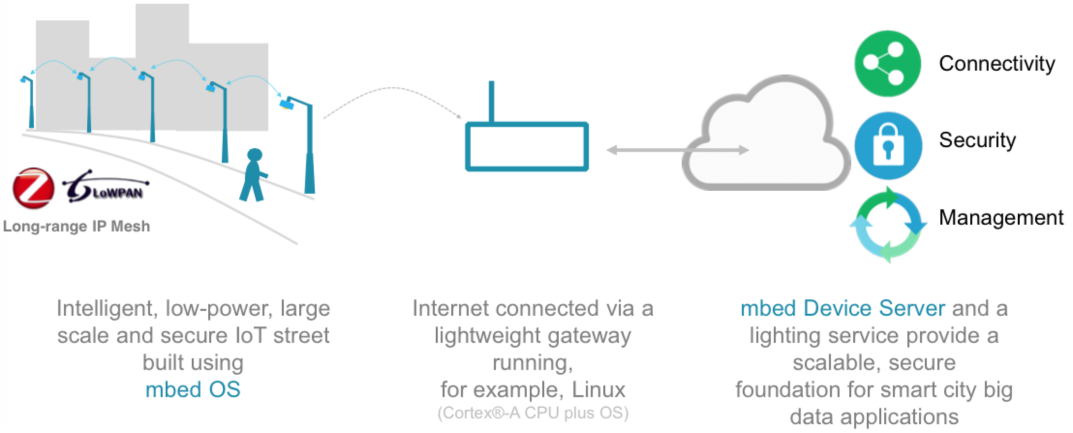
The layers of the mbed server software ... your apps sit at the top

But what's it all for? A good question. In fact, ask a room of embedded engineering execs for examples of sensors and killer apps for the Internet of Things, and you may well end up sitting in a brief awkward silence. The fact is, we're still very much in the "build" part of "build it and they will come".

ARM is working with IBM on its Smart Cities. This means street lights being able to sense, using Wi-Fi or Bluetooth, how many people are nearby and adjusting their lighting accordingly, via a system running mbed Device Server. This can be more fine-grained that relying on primitive motion detectors.

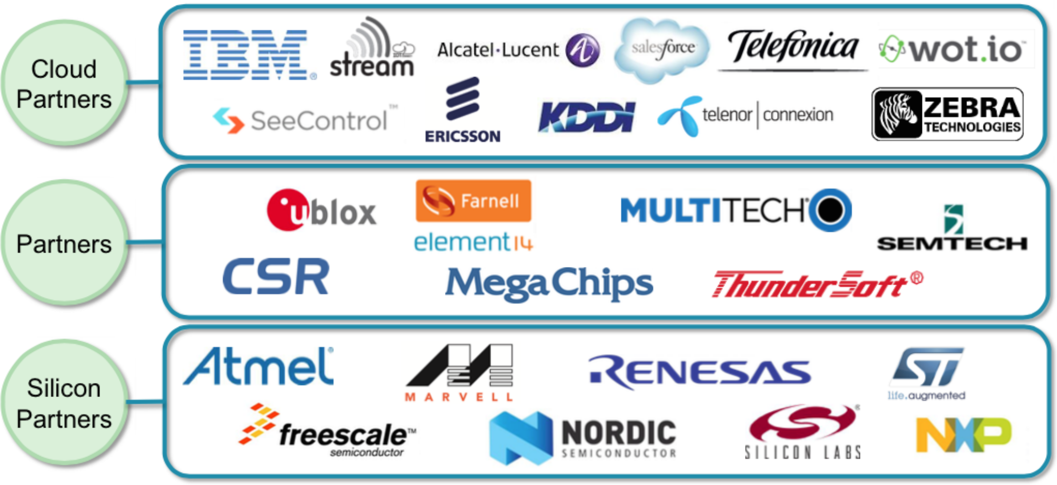
A home could have movement sensors on kitchen cupboard doors to detect when kids are swiping cookies, thinking mum or dad isn't looking – but instead an alert goes off on the parent's smartphone, and the child's tablet is locked as punishment. Why not just put a lock on the cupboard? Well, maybe little Luke or Lisa deserves a cookie from time to time, so the parent tells the phone it's allowable.

Maybe, you'd simply like to control your oven from your smartwatch. ARM's hoping lots of people will start to solve problems in areas where there's sufficient levels of demand to sell products and make money. Because that means ARM can make some money from it, too.


6LoWPAN
Long-range IP Mesh
Connectivity
Security
Management
Intelligent, low-power, large scale and secure IoT street built using mbed OS
Internet connected via a lightweight gateway running, for example, Linux (Cortex®-A CPU plus OS)
mbed Device Server and a lighting service provide a scalable, secure foundation for smart city big data applications

So how's this thing work ... street lights that know who's there (click to enlarge)

ARM says it's got a load of companies on board already in the new-look mbed project – from IBM doing the aforementioned Smart Cities to system-on-chip manufacturers taking the Cortex-M core blueprints and building peripherals around them in silicon. These manufacturers are expected to support the mbed OS, from the chips to the boards.



Who's on board ... IBM to Ericsson, Atmel to Marvell, etc (click to enlarge)

The mbed OS should be ready for ARM's partners to use by Q4 2014, we're told; don't expect any devices using the software until 2015. Mike Muller suggested it will take about three years before vendors start to see real take up of mbed-powered IoT things, if the platform takes off. The mbed Device Server is already in production, apparently.

The software is described as free, and partially open source, but ... there's always a but. The mbed OS is free to use, and a lot of it will eventually be open source, we're told. But there will be binary blobs in the firmware. Some of this will be closely guarded, closed-source drivers for the SoCs, developed by the chip manufacturers; some of it is for security; some of it is so that features remain closed source until they're finished.

For example, it should be possible to update the microcontroller firmware over the air, so you'll want those upgrades to be cryptographically signed so that attackers can't upload arbitrary malicious code to devices. That means including a public key in the flash.

Another part of it is what some may describe as control, but ARM describes it as consistency. You can tweak and improve the open-source parts of the mbed stack, but ARM would like particular components to remain consistent and fully compatible throughout the platform – particularly the security and radio hardware layers, and the code that brings up a device.

When an mbed OS SoC starts up, it may be possible for the device to fetch and install firmware that best suits its surroundings; to do this, the SoC has to initialize into a secure state running instructions it can trust; there's little to gain from securely update an already compromised device.

Again, this is where binary components come into play. ARM partners will get access to the source, however.

Security, we're told, is included from the ground up, rather than as an afterthought or premium feature. The kernel has been designed to run the on-chip software in a "cryptobox", a sort-of sandbox that attempts to stop code compromising the device. The Cortex-M family doesn't feature memory management units, nor TrustZone, so securing programs must be done in software.

Until the more curious-minded start jailbreaking $5 sensor SoCs to run completely arbitrary code, the binary part of the stack allows ARM to keep a firm hand on the tiller, to stop the project fragmenting and taking the team back to square one.

"Open source makes it easier to improve, and easier for people to find and fix bugs," ARM's Shelby added.

Meanwhile, the mbed Device Server side is free to use and develop on, but a license must be obtained to use it commercially. This is ARM, after all; licensing technology is its bread and butter. ®

Tips and corrections

20 Comments

More from The Register

Microsoft cuts Facebook Messenger, Google Talk from Outlook.com

Skype, anybody?

15 Comments

EU: Explain your tax affairs. Google, Amazon, Facebook: Mmm... nah

Toothless Eurocrats wail and gnash gums over big biz invite snub

35 Comments

EU, Google, Facebook, Twitter, Microsoft: We'll fight terrorists... with WORKSHOPS

EU has dinner with Big Tech before debating data protection

8 Comments

Facebook, Google execs cough to their biggest privacy

Google joins Bluetooth snoop pals with iBeacon rival tech

Searching to destroy ... Bing? Facebook JILTS Microsoft

blunders

RSA 2015 Beacon for Zuck & Co, not hard to guess Google's

9 Comments

Eddystone

Apple isn't the only one that wants to track your smartphone

9 Comments

Relationship status: SINGLE. Boom!

26 Comments

The Register

Biting the hand that feeds IT © 1998–2015

Independent news, views, opinions and reviews on the latest in the IT industry. Offices in London, Edinburgh, San Francisco and Sydney.

About us

Privacy

Company info

Advertise with us

Syndication

Send us news tips

More content

Newsletters

Top 20 stories

Week’s headlines

Archive

eBooks

Webcasts