



Automate over-the-air (OTA) software updates across your embedded Linux device fleet

Deliver bug fixes, patch security vulnerabilities, and rollout new features
to field-deployed embedded Linux devices (Internet of Things)
with a robust open source OTA updater

Automated over-the-air updates

Mender is an open source project (Apache 2.0) with a client/server architecture to manage the deployments of OTA software updates across a fleet of embedded Linux devices. Mender provides real-time device health reporting and removes the need to develop a homegrown updates management tool, which can take two to twelve months to build for a full time employee (FTE), as well as the added costs in maintaining it.

Mender addresses the following:

Rapid security patching

By automating OTA updates, software vulnerabilities are timely patched to reduce the attack surface of identified exposures, limiting exposure windows for hackers.

Extensible

Support evolving requirements and work in conjunction with other tools in the stack. Mender works the way you do today without requiring developers to revamp their workflow.

Maximize device uptime in the field

A robust update process is critical for embedded devices. They can lose power and connectivity at any time during the update process. Mender implements atomic updates with a dual A/B rootfs partition layout so your device will always have a working environment.

Real-time visibility into device fleet

Mender's management server provides real-time reporting of the entire fleet of devices with configurable reports and dashboards.

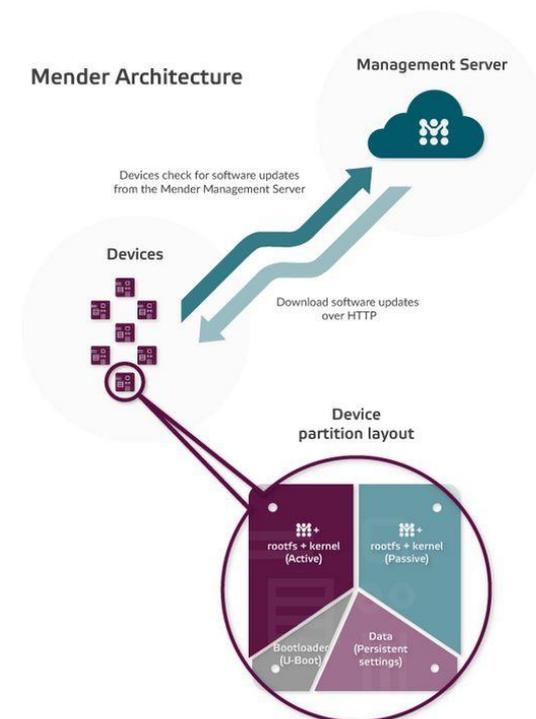
Intuitive and easy-to-use

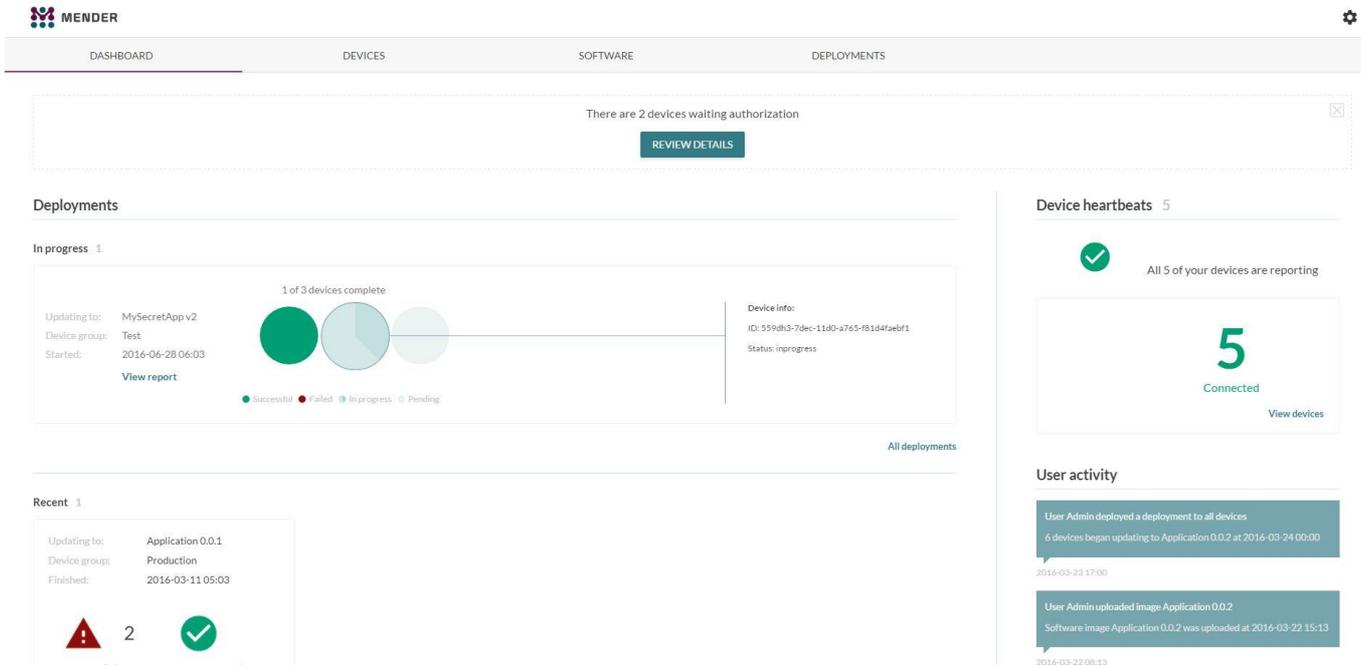
Mender's intuitive user-interface can be used by a variety of team members without the need for a specialized technical skillset.

Retire that USB stick

Automation of over-the-air updates removes the manual labor to update each device individually and eliminates the high cost of sending a technician to the field to repair any device outage.

Architecture





The screenshot shows the Mender dashboard with a navigation bar (Dashboard, Devices, Software, Deployments) and a settings icon. A notification at the top states "There are 2 devices waiting authorization" with a "REVIEW DETAILS" button. The main content is divided into three sections:

- Deployments:**
 - In progress 1:** A deployment for "MySecretApp v2" to the "Test" group, started on 2016-06-28 06:03. A progress indicator shows "1 of 3 devices complete". A legend indicates: Successful (green), Failed (red), In progress (blue), Pending (grey). A "View report" link is present. Device info: ID: 559d93-7dec-11d0-a765-f81d4faebf1, Status: inprogress.
 - Recent 1:** A deployment for "Application 0.0.1" to the "Production" group, finished on 2016-03-11 05:03. It shows 2 failed attempts (red triangles) and 1 successful attempt (green checkmark).
- Device heartbeats 5:** A green checkmark icon and the text "All 5 of your devices are reporting". A large green "5" is displayed with "Connected" below it and a "View devices" link.
- User activity:**
 - 2016-03-23 17:00: "User Admin deployed a deployment to all devices. 6 devices began updating to Application 0.0.2 at 2016-03-24 00:00".
 - 2016-03-22 08:13: "User Admin uploaded image Application 0.0.2. Software image Application 0.0.2 was uploaded at 2016-03-22 15:13".

Features

- Rootfs rollback support
- Device status at a glance
- Group-based deployments for controlled rollout management based by customer location, or any other way
- Yocto integration with the meta-mender layer
- Secure TLS communication between client/server
- Intuitive management UI
- Signing and verification of image artifacts for authenticity *

*Planned feature

Operating system support

Mender currently supports Linux-based devices that use U-Boot. Mender provides a meta layer for the Yocto Project for very easy integration with Yocto-based distributions. While it is possible to compile and install Mender independently, we have optimized the installation experience for those who build their Linux images using the Yocto Project.

Board support

Mender uses the BeagleBone Black as the reference board – it is in our continuous integration process. Since Mender integrates with existing Yocto Project environments, it is easy to extend board support by following the Mender documentation: <https://docs.mender.io/Devices>

Device storage

Client binaries, which are written in Go, are approximately 7 MiB in size. Mender supports eMMC and SD Card-based storage at the moment. Support of raw flashes are planned. Please contact us if you are interested in raw flash support.

Boot loader support

Mender uses the U-Boot feature `CONFIG_BOOTCOUNT_ENV` to support atomic deployments and rollback, which is available in U-Boot v2014.01 or newer. If you have an older version of U-Boot, it should be straightforward to backport or implement this feature. Please contact us for any guidance.