info@apriorit.com
www.apriorit.com

Case study

# Developing a Custom MDM Solution with Enhanced Data Security

Our client wanted to create an enterprise MDM solution for securely managing Android tablets. The Apriorit team developed a custom MDM solution with enhanced data security and an admin panel for centralized management of all tablets connected to the client's system. This allowed our client to deliver their services to Android users and maintain the needed level of data security.

## The client

Our client is a US-based company providing communication services to organizations with strict data security requirements, including corporations and federal institutions. They provide their customers with tools for securely exchanging media as well as holding secure audio and video calls.

## The challenge

The client needed a tamper-proof MDM solution with enhanced overall data security and limited device management capabilities on the part of the end user. They also requested a centralized admin panel for managing and auditing all tablets with the MDM solution installed.

Additionally, our client decided to implement a custom store-like service with a rich selection of safe and useful media content. This would help them expand the range of services they could offer their end users.

## Our solution

Apriorit specialists analyzed possible scenarios for developing a custom MDM solution that would meet all of the client's requirements. We discovered that the only way to build the solution our client wanted is by changing device firmware. Therefore, we offered to customize tablet firmware in a way that would allow our client to:

- Prevent users from rebooting, resetting, loading in safe mode, or turning off the device
- Disable user access to device settings
- Prevent users from establishing any unauthorized internet connections
- Provide admins with access to managed devices via the admin panel

In addition, our specialists suggested establishing an activity monitoring system for timely detection of suspicious actions with tablets.

| Apriorit Team |
| --- |
| • Business analyst |
| • Project manager |
| • UI/UX designer |
| • Quality assurance specialists |
| • Frontend developers |
| • Backend developers |
| • Android developers |

| Key Technologies | |
| --- | --- |
| • C# 7/8 | • Retrofit2 |
| • ASP.NET | • Room |
| • Kotlin 1.3 | • Firebase Cloud Messaging |
| • Android | • Dagger 2 |
| • SignalR | • Timber |
| • OData | |
| • Swagger | |

## Our approach

Before building an MDM solution with strong data protection, we elicited project requirements, negotiated the most effective communication scenarios, and prepared the preliminary design and architecture for the solution. After that, we split the overall scope of work between two main stages:

<table>
<tr><th colspan="3">Project Stages</th></tr>
<tr><td>1</td><td colspan="2">2</td></tr>
<tr><td rowspan="2">MDM solution development</td><td colspan="2">Admin panel implementation</td></tr>
<tr><td>Backend development</td><td>Frontend development</td></tr>
</table>

One of our client's key requests was to ensure strong protection of all application data and enable strict device management limitations for end users. To fulfill this request, we offered to implement a set of sophisticated cybersecurity measures.

### Enhancing data security with suspicious activity monitoring

At Apriorit, we understand the value of cybersecurity because we help our clients strengthen the security of their products for more than 18 years. We made sure to address data security in our client's MDM solution on several levels:

- **Firmware customization** to prevent end users from making unwanted device manipulations, including resetting to factory defaults and booting to safe mode
- **Limiting device configuration capabilities** to ensure end users can't change device configurations
- **Suspicious activity monitoring** to allow admins of the client's system to receive notifications when an end user attempts to perform restricted actions on a monitored device; notifications can be tied to different triggers associated with suspicious data or device manipulations

## Building an MDM solution

We started by developing an MDM solution for Android tablets. First, we defined the set of features to be implemented in this solution:

| Initial MDM Features | |
|---|---|
| Enrollment | Logging |
| Remote control | Tablet monitoring |
| Navigation restrictions | UI customization |

Each of these features plays a crucial part in the MDM solution's performance:

- **Enrollment** is responsible for registering the device in and removing it from the system.
- **Remote control** enables system admins to manage each device remotely.
- **Navigation restrictions** enable the application of various device use policies, blocking activities, and configuration changes specified in such policies.
- **Logging** gathers device activity data, which can be helpful for both fixing bugs and analyzing suspicious events.

- **Tablet monitoring** provides the means for continuously monitoring a tablet's critical parameters, such as battery state and built-in storage capacity.
- **User interface customization** allows for removing restricted options and capabilities from the default tablet toolbar.

Later on in the project, we integrated all these features into the admin panel's back end.

While we worked on the core functionality of the MDM solution, the client approached us with an additional request — to add a custom marketplace similar to the Google Play Store. They planned to fill this marketplace with different types of media content including e-books, music, and applications.

To implement this service, we had to revise both the budget and project deadlines. After negotiating the details with the client, we added this new feature to the scope of work and successfully implemented it.

## Implementing the admin panel

When we finished working on the MDM solution features, our team moved to building the admin panel that would enable remote management of all devices added to the system. At this stage, our developers focused on three major tasks:

1. Integrate the admin panel with the device
2. Build the admin panel back end
3. Develop the admin panel front end

As the front end and back end of this solution are highly interdependent, we implemented them in parallel.

Based on the design the client approved at the project discovery phase, we split the admin panel implementation process into a series of features:

| Key Admin Panel Features | |
|---|---|
| Users | Devices |
| Reporting | Notifications |
| Dashboard | Market management |

We started by implementing the functionality needed for managing **users** and **devices** registered in the system.

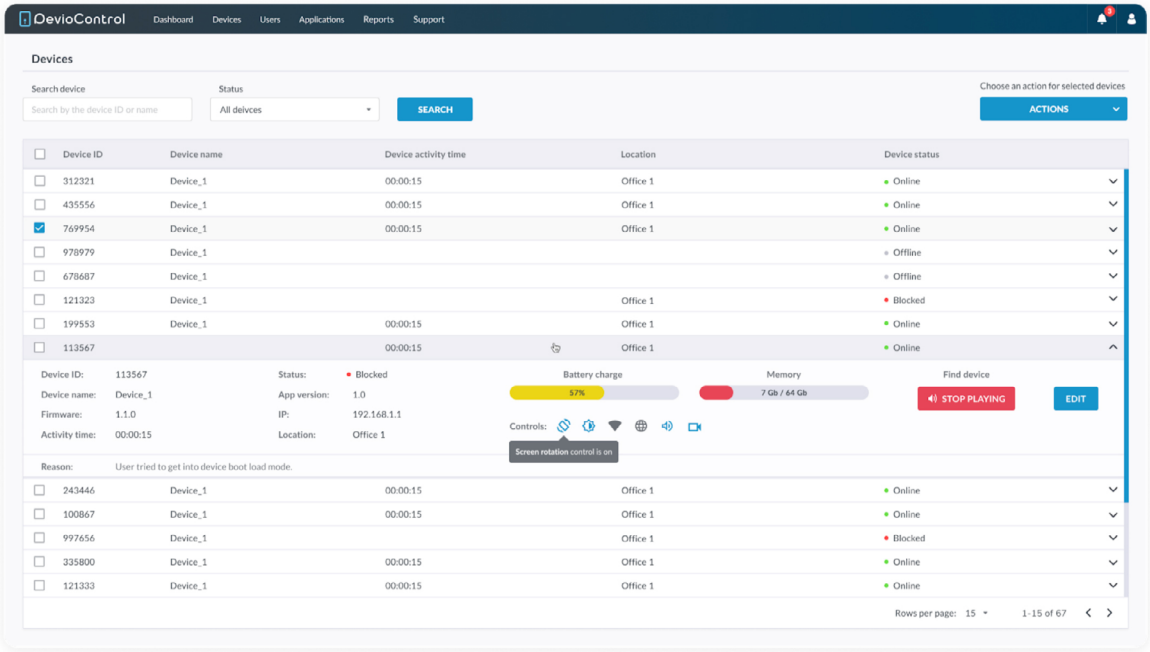


*Figure 1. Devices view*

**Reporting** and **notification** functionalities would provide system admins with all the information needed for effective device management.

The **dashboard** feature provides admins with easy-to-understand metrics and visuals on different parameters.

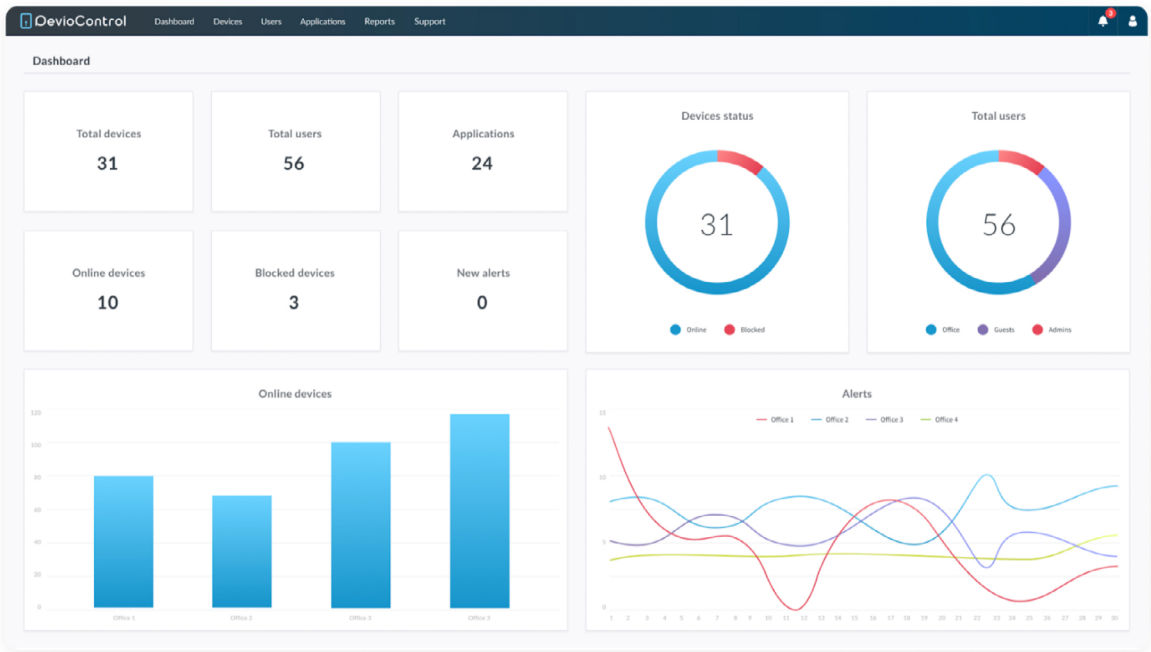


*Figure 2. Dashboard view*

Finally, the **Market management** feature provides all tools and capabilities needed for managing content that end users can access through the custom Market app on their tablets.

As our team worked on these features, we also continued improving integration of the MDM tool and the admin panel. As we had no access to the firmware source code, this created additional

challenges during the project. Thankfully, the professionalism and persistence of our developers helped the team successfully overcome these challenges.

## Challenges and solutions

Throughout the project, the Apriorit team had to handle two major challenges:

**1. Ensuring smooth integration without access to tablet firmware source code**

This project relies on custom tablets manufactured by a third-party vendor.

Unfortunately, the client's vendor refused to provide us with access to the tablet's firmware source code and insisted on implementing all firmware changes themselves. Furthermore, the speed and quality of the vendor's work didn't always meet our expectations.

We overcame this challenge by configuring an Android 9.0 environment that mimicked the known parameters of the custom firmware. Based on our analysis of this environment and its behavior, we composed highly detailed requests for the third-party vendor's team to implement.

**2. Addressing issues caused by the Firebase Cloud Messaging protocol**

Based on the initial information we had about the project, we decided to implement device–back end communication using the Firebase Cloud Messaging protocol. However, when we started implementing this solution, it became obvious that the Firebase Cloud Messaging protocol wouldn't fit the needs of the product.

Our developers dealt with multiple integration issues, such as undelivered device commands and failed Firebase connection attempts. While we have found quick fixes to these issues, such as implementing a command delivery confirmation mechanism, these measures are temporary.

To ensure flawless performance of the MDM solution in the future, we're now evaluating other communication protocols. We plan to switch to Azure IoT or Ably with the release of a new product version.

## The impact

The MDM solution developed by the Apriorit team allowed our client to expand their services and maintain the required level of data security. The client has already released the first 5,000 tablets and distributed the devices among their customers. Meanwhile, our team continues working on both improving the current solution and implementing new features.

**Want to build a custom MDM solution with strong data protection and flawless performance? Delegate this task to the Apriorit team!**