GUIDE

BlueCedar

Solve Microsoft Intune Deployment Challenges with Post-Development Mobile App Modifications

A Best Practices Guide

BIAC Representing BlueCedar in Canada.
biacbroadband.ca (866) 941-5119 Ext. 120

bluecedar.com

Once app development is complete, the need to make modifications to a mobile app is quite normal, if not expected. Ensuring that mobile apps on unenrolled devices without mobile device management (MDM) controls can access on-premises resources, which are protected by corporate firewalls, and have app-level security controls are common preconditions for making apps available to end users such as employees, contractors and consultants. The responsibility for such post-development app modifications typically falls to DevOps and IT Operations teams, which are responsible for deploying mobile apps to end users.

Microsoft Intune is a cloud-based service that provides mobile application management (MAM) and MDM. Enabling mobile apps, whether developed by internal development teams or third parties, with Microsoft Intune MAM provides companies with the app-level security controls they need to permit use of personal devices that do not have MDM for work, as corporate data can be protected and kept separate from personal data.

At first glance, performing app modifications to incorporate Microsoft Intune MAM controls into already developed apps potentially introduce risks and other down-stream issues into code. To avoid these risks, this guide, intended for DevOps and IT Operations teams, will outline the recommended best practices for performing post-development modifications to mobile apps in Microsoft environments.

Challenges for Post-Development App Modifications in Microsoft Environments

What drives the need for making modifications to a mobile app after development is complete? There may be underlying infrastructure, security, performance, or analytics functionality overlooked during the initial code build. But whatever the reason, modifications can add both complexity and expense. Forrester estimates the average amount spent developing a typical customer mobile app is just 35% of the actual two-year cost[1].

If you want to deploy apps with Microsoft Intune MAM, some specific challenges you may face for post-development app modifications include:

1. You need to connect apps to on-premises resources from unenrolled devices

End users, such as employees, contractors, and consultants, are more productive and efficient when you make it easy for them to access corporate "behind-the-firewall" apps and work from any device. However, there is no convenient way for users of Microsoft Intune-enabled apps on unenrolled devices to access on-premises resources protected by corporate firewalls. Launching a separate VPN app or a device-level VPN to access remote resources will disrupt the user experience and decrease usage of these Intune apps. It also intermingles corporate and personal traffic as all traffic is routed through the corporate VPN.

2. You are deploying a third-party developed app but can't enable all Intune controls because you don't have the source code

If you want to deploy an app developed by a third-party vendor and want to add Intune MAM capabilities, you will need to modify the code. While Microsoft provides app wrapping technology to add Intune MAM to apps without having to write code, it doesn't enable all the app protection policies that are available when using the Intune SDK. The logical step to take would be to procure or license the actual source code of the app and write code to integrate the Intune SDK. But this is not always feasible. Significant, prohibitive licensing fees are often involved. And if you can acquire and pay the license fees, a qualified technical team to perform the modifications may not be readily available.

3. You are deploying internally developed apps and want to enable Intune, but you have no access to qualified developers

If you have the source code to an internally developed app, you will need a development team to make any changes or additions. The development team needed to make those modifications may require different skill sets than the initial development team, skills that may be in short supply. This new development team needs to become familiar with the app's source code and Intune SDKs. The SDKs are also updated on a regular basis, so the team will also be responsible for ongoing security-related maintenance of the app, including use of any new features introduced by Microsoft.

4. You are switching to Microsoft Endpoint Manager from another UEM and need to support both environments for a period of time

Maintaining two Unified Endpoint Management (UEM) solutions is always going to be a challenge as it encompasses many facets. There is a big support burden for the deployment and help-desk teams, and teams need training on legacy and emerging features and capabilities. Then, there are the security issues revolving around maintaining multiple versions of apps. Finally, there are likely licensing and distribution considerations—all of the preceding results in increased cost to the company.

5. Your employer is not using MAM mode, but should

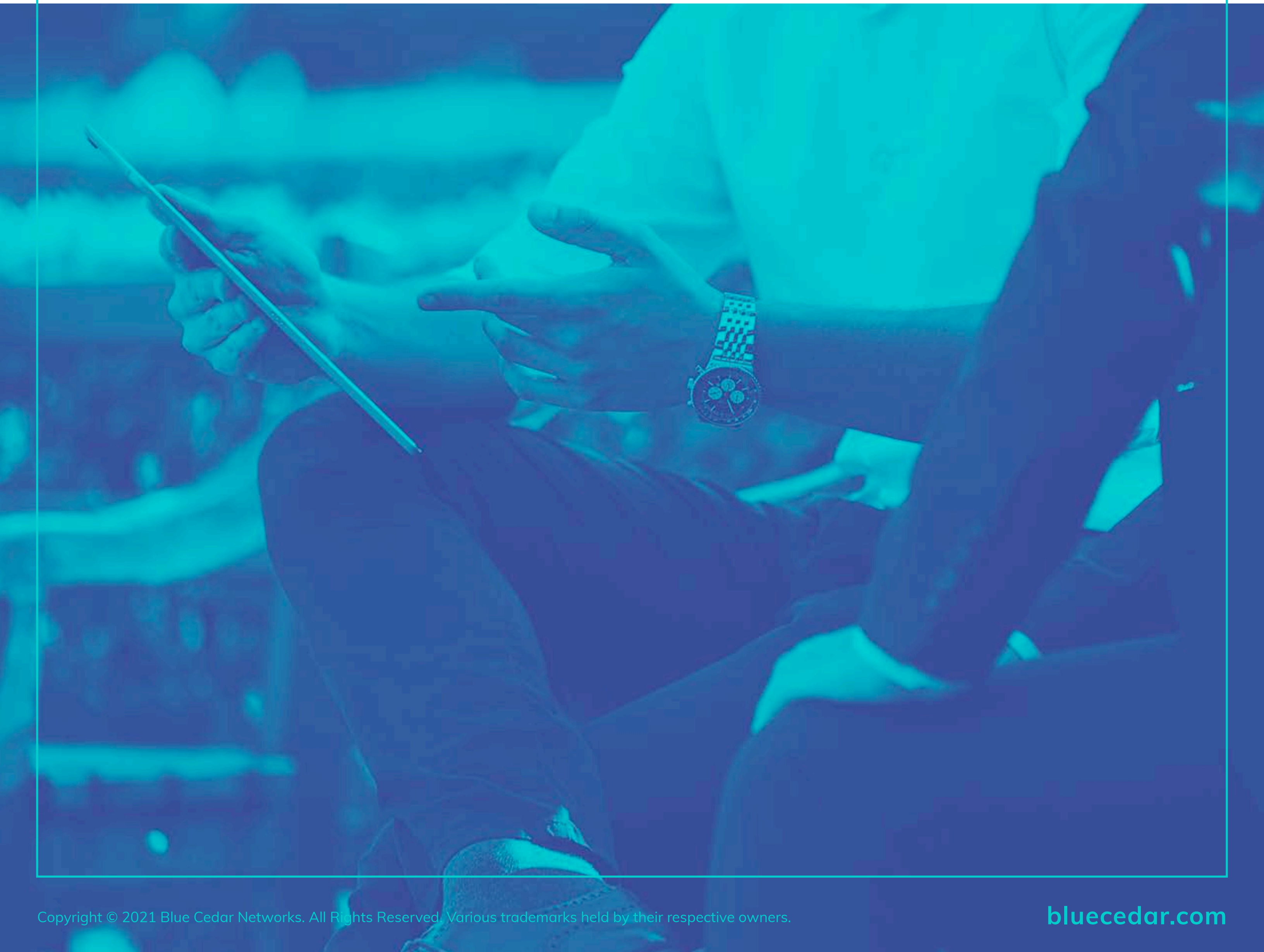
MAM provides granular controls (at the application level) that enable companies to manage and secure app data, even on devices without MDM. For various reasons, including complexity, lack of knowledge or training, lack of a viable method for remote access, or lack of prioritization, a given company might not use Intune MAM.

[1] https://www.computerworld.com/article/2501477/chief-mobile-officer--a-job-title-now-timely-.html

Despite all these challenges, there is significant upside for companies making modifications post-development. The primary is control: DevOps and IT Operations teams can make the changes they need without hindrance. Second, there is speed. These teams can make changes when they want, keeping their business agile. When you consider that apps come from many different sources—internally developed, third parties, citizen developers, etc.,—control and nimbleness are essential to ensure that DevOps and IT Operations can rapidly deploy the apps that LoBs want with the controls that companies need.

Post-development app modifications do not happen in a silo. Other deployment activities must be completed before modified apps can be made available to end-users. Once an app is changed, it has to go through the rest of the deployment work cycle—signing, distribution, and testing. When there are many apps, each with a potentially different modification, the processes can get very complicated. Using workflows helps to streamline all activities and reduces risk by providing repeatable and tested procedures.

But what can companies do to increase their agility and speed, provide the required app functionality, reduce other operational risks and ensure the likelihood of success?



Best Practices for Post-Build App Modifications

Given the complexity mentioned above, potential resource constraints and added cost, the following best practices can maximize your mobile app deployment outcomes while minimizing friction between teams and organizations at your company. Make sure your team considers the following:

1. Use an in-app VPN and make sure it is optimized for mobile environments. Companies moving to Intune have large volumes of on-premises data that can enable higher organizational productivity but only if all end users have secure access to it from mobile apps on any device. An in-app VPN ensures secure access to this data from mobile apps even when the mobile device does not have mobile device management (MDM) controls. Use an in-app VPN that accommodates the unique requirements of mobile devices, such as power conservation and bandwidth optimization, by using transient connections rather than long-lived VPN connections.

2. Use a no-code integration service and make sure it supports the type of functionality and app frameworks that you need integrated. A no-code integration service makes it easy to add new functionality to mobile apps without requiring source code access or writing code. Use a no-code integration service that allows you to add mobile application management (MAM), in-app VPN, or authentication functionality to any app, along with the ability to mix any combination of the functionality. For example, integrate MAM, in-app VPN, and authentication, or just integrate MAM. Additionally, ensure the no-code integration service you select can be incorporated into deployment workflows to enable a seamless path to modify already developed mobile apps and distribute them to end-users.

3. Codify activities into deployment workflows to help ensure reliability and repeatability. Workflows also simplify and ease the rest of the mobile app deployment flow such as signing, distribution, etc. When building workflows, separate the pilot from production deployments, ensuring you get things right before rolling it out to everyone.

4. Import from code repositories to make the process as seamless as possible. Ideally, you should use tools that notify you there is a new version of that app and provide an option to automatically import apps into workflows.

5. Use orchestration to coordinate the different workflow activities across people and services with the ultimate goal to get the app out to end users. Post-development modifications are just one use case for an orchestration platform. For example, companies typically have multiple teams working on different parts of the same project: one is responsible for app signing, another for getting the app published to a public store, while others focus on working through compliance and security. Orchestration replaces the manual coordination of deployment efforts across teams and technologies, enabling efficient and error-free workflows. It enforces an agreed upon sequence of deployment activities, automates tasks where appropriate, and coordinates notification when manual intervention is required to progress a workflow. Orchestration eliminates deployment delays while supporting compliance with security policies and regulations, and provides the recipe for ensuring rigor for app deployments, especially when there are many workflows with unique recipes or activities.

6. Use notifications so teams and departments are instantly informed about critical events in the post-build phase. For example, if your no-code integration failed, a warning could alert you and point you to a report that explains why the integration failed. Notifications are also useful for alerting teams to status changes, perhaps notifying them that an app is ready to be signed or is ready to be distributed. Using notifications via email, SMS, or chat-collaboration, is a highly efficient way to communicate across teams and organizations within a company.

7. Use the audit trail to keep a record of all changes, including a historical record of all app versions. Audit trails are essential for compliance and suitable for reverting to the last good version of an app. They are also useful if the SDKs being integrated break the app during the modification phase.

Benefits

Security with a better end-user experience

In-app VPNs work well in a Bring Your Own Device (BYOD) context, where devices will not have MDM controls as network traffic from different apps is kept separate, apps can't exchange data, and the use of personal apps is not impacted. Discrete in-app VPN connections enable multiple security access levels from an end user device, as different apps can connect via different VPN configurations and servers. In contrast, a shared device-level VPN only supports a single security access level. An in-app VPN also facilitates a better user experience as end users do not need to remember to turn the device-level VPN on and off, as the VPN connection is automatically established when the app is launched.

Accelerate mobile app deployments with lower costs

Faster app deployment is a clear benefit and win, as it will increase overall business velocity. Taking advantage of workflows, deployment services such as no-code integration, and abstraction of deployment activities from the underlying technologies—components that should be part of an Orchestration Platform—will also reduce costs, as repeatable processes help guarantee meeting all deployment checks while minimizing team resources.

Ensure consistent use of workflows

Employing consistent workflows leads to enhanced productivity, reduced risk of non-compliance, enhanced security, reduced reliance on IT and developer teams, and overall lower support and maintenance costs.

Deliver comprehensive visibility to enable and enforce controls

Real-time visibility into app deployments provides a clear window into the possible issues present during the deployment process. Retroactive visibility through an audit trail provides evidence needed to demonstrate compliance. Visibility also paves the way to greater controls in the deployment process, thus minimizing errors and security risks.

BlueCedar

The Blue Cedar Platform is transforming mobile app deployment by helping Fortune 500 companies, governments and independent software vendors orchestrate all app modification, security, compliance, and release activities in unified deployment pipelines. The Platform includes a no-code integration service that adds new functionality to mobile apps without requiring source code access or writing code. Blue Cedar integrates with popular tools and systems used in mobile app deployment, including GitHub, GitLab, Microsoft Endpoint Manager, BlackBerry UEM, Digital.ai, Google Play and the Apple Custom Store. Founded in 2016, Blue Cedar is funded by leading venture capital firms and is headquartered in San Francisco. For more information, visit www.bluecedar.com.

325 Pacific Avenue, San Francisco, CA 94111
info@bluecedar.com / bluecedar.com

02022021