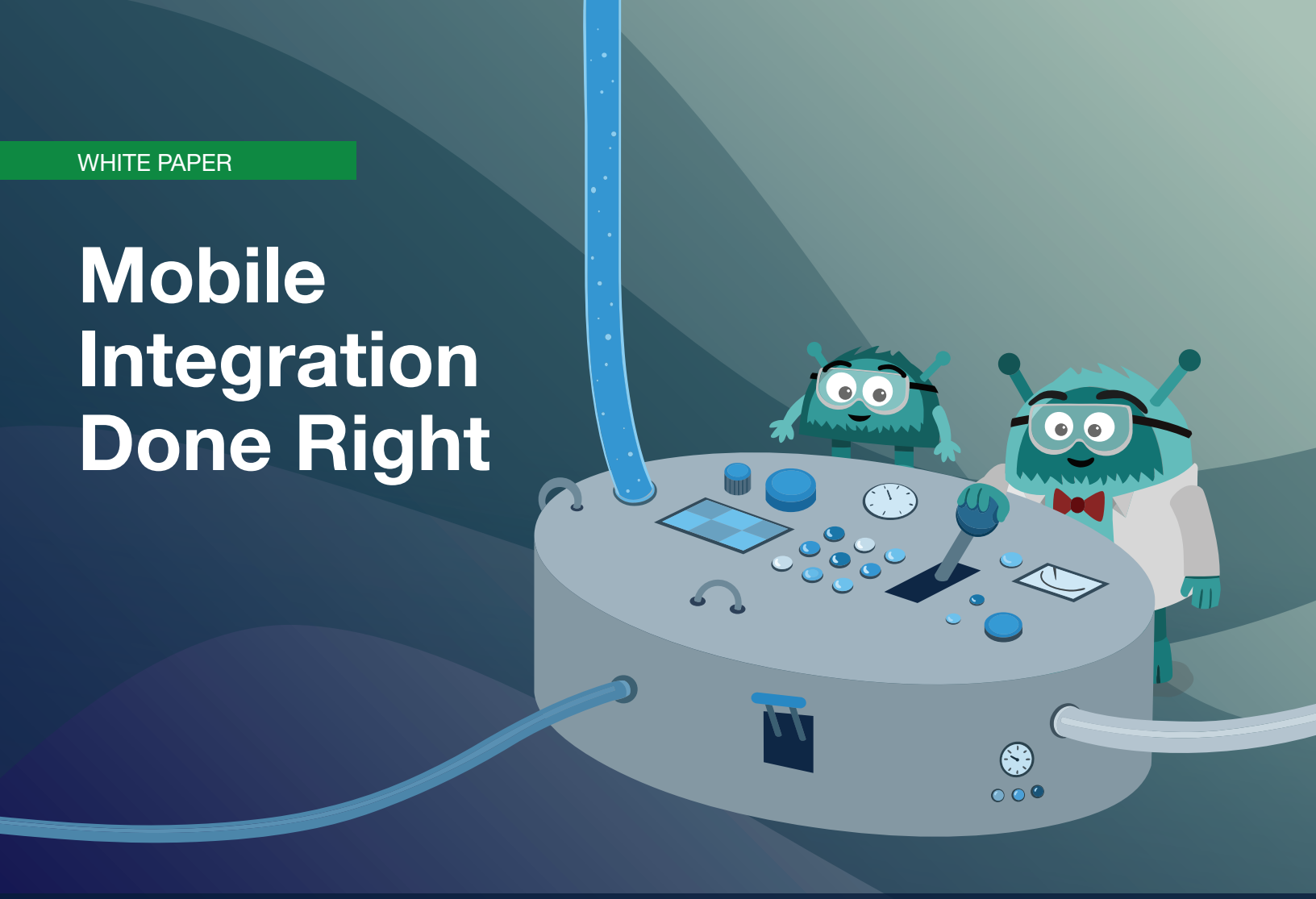
WHITE PAPER

Mobile Integration Done Right



Fusion automated mobile app integration, saving you time, money and more

There are millions of consumer and enterprise mobile apps. There are billions of smartphones, tablets and IoT devices. Mission-critical services are being brought to the mobile environment through apps at an increasing rate and across a widening diversity of platforms.

The sticking point is integrating the new mission critical mobile experience with other services and solutions. Because of increased demand and limited resources, organizations can't develop, secure, integrate and manage all the apps they need to. Most integration solutions take too long. They are expensive and require significant development resources. Even when done right, they can impact stability, functionality and performance of the app they're intended to improve.

There is a better way.

appdome

Mobile SDKs

Mobile SDKs aren't automatic. They're useful to the knowledgeable enterprise with lots of resources. However, SDKs still require platform-by-platform builds, separate releases by version and multiple QA cycles.

Mobile OS APIs

Mobile OSs are gaining new app management capabilities but it is early days and often not suitable for large-scale deployments where consistency and security are important.

Integrating Security, Mobility Management and Beyond

Mobile app developers, mobility professionals and end users all want awesome apps with seamless integration into services and solutions they trust. When presented with the challenge of integrating mobile apps developed using multiple programming languages on multiple mobile development platforms for multiple platforms with EMM, MAM, security and other capabilities, there are significant challenges. At best, integration is time consuming and resource intensive. The reality is Multiple QA and release cycles create nightmare scenarios for the mobile developer that wants to serve an enterprise customer. A mobile developer or Independent Software Vendor (ISV) would typically have to choose from three flawed integration approaches.

SDKs

Software development kits (SDKs) aid the development process and are offered by every major EMM, MDM, and MAM vendor. But, SDKs still require a heavy development effort to implement, mange and maintain. SDKs are vendor-specific, solution-specific, and platform-specific. So, SDKs solve a tiny fraction of the overall mobile integration challenge. SDKs may make a discrete integration project easier but the overall integration effort across multiple projects, versions, platforms, and use cases remains heavy.

SDKs, by themselves, aren't automatic. Holding everything else constant, SDKs replace direct source code integration but impose a "new step" in the integration process. SDKs require new expertise, one tailored to the specific SDK, and still require platform-by-platform builds, releases and QA.

Mobile OS Management APIs

Mobile operating systems have started to incorporate mobile management APIs into their platforms. This allows mobile developers and independent software vendors (ISVs) to use the management controls inside the Mobile OS to manage devices and apps. The biggest advantage of these built in features is that the mobile developer or ISV need not compile a version of the app with a specific SDK to allow users to mange the app. The biggest dis-advantage is that the OS-based feature set isn't universal across platforms. One OS might support one set of features while another OS supports a different or lesser set. This divergence leaves some users and some enterprises without access to critical features needed in their environment. In many cases, the missing features force the app to fail security requirements, which hinders deployment and adoption of the app. The mobile OS approach also forces the device to be enrolled in a single EMM system. In an increasingly mobile workplace, where contracting is commonplace, this is often impractical for large-scale deployments.

App Wrapping

Wrappers are quickly becoming a thing of the past. Initially thought to make managing apps easier, they obscure the underlying application code, making debugging and distinguishing between wrapper errors and native application errors nightmarish.

App Wrapping

Application wrapping promised more rapid mobile application integration than SDKs. But, like SDKs, application wrappers were platform-specific, solution-specific, and lacked standards.

Wrapping also spawned new problems such as imposing functionality limitations, often resulting in unpredictable behavior. Wrappers rely on intercepting or modifying application layer APIs. Wrapping stand between the native application functions and the EMM or MAM and can actually collide with applications. For example, when a mobile application uses frameworks or is built using MADPs (Mobile Application Development Platforms), basic EMM functions such as containerizing files becomes an impossible challenge for wrappers. The wrapper is sensitive to loading order, framework usage and application logic. Wrappers often cause applications and frameworks to fail. Interacting with a huge and ever-growing number of possible APIs leaves a lot of room for error. It also leaves a lot of functionally walled off by the wrapper, making wrapping undesirable for most developers. Wrapping also poses problems for commercial apps. ISVs typically don't support wrapping as it interferes the apps native operations.

The following chart illustrates an abbreviated, high-level comparison of wrapping and Appdome's Fusion.

App Wrapping	Fusion
Limited by platform/vendor/version use case	Open across all platforms/vendors/versions use cases
Limited to specific app functions	No dependencies on specific functions or vendors
Limited to only one SDK per app	Multiple services can be fused to the same app
Time and resources consuming (weeks or months)	Minutes to implement; minutes to re-implement in a new app version
Manual process	Automated process with an intuitive, simple, powerful cloud-based platform
Restricted visibility into overall app behavior	Complete visibility into how the app operates
Requires additional frameworks and libraries	Fully integrated into the binary without the need for additional frameworks to operate and libraries
Unsupported on AppStore or GooglePlay	Fully supported on AppStore, GooglePlay and enterprise stores
Leverage "private" APIs to "proxy" and "hook" high level API calls within the app	Does not leverage any "private" APIs
Dynamically load code into an app at run-time so as to pass Apple's static inspection	Does not rely on dynamically "injecting" or "hiding" code from Apple's static tools
Leverage special linkage (non-PIE) causes unsuccessful submission to the App Store. Also, does not support new Apple technologies and chipsets (such as 64bit)	Appdome's fusion layer is generic and suitable to all binary types, linkage stacks and run-time environments out of the box
Modify both app and the Apple ID bundle during signing	Built-in signing workflow on the Appdome platform that ensures each app is App Store ready
Add additional permissions (entitlements) that are not in the original provisioning profile, causing Apple to reject the submission	Does not add any additional permissions (entitlements) to app
Does not allow customization of apps	Easily allow multiple apps to be customized during the Fusion process

Fusion — Your app the way you meant it to be.

With Appdome, your application remains intact and unaltered. An app calls the Fusion code as the first element of the binary. Every OS interaction such as file IO and networking gets processed through the Fusion code. The Fusion code can be reconfigured, and even turned off to make debugging easier.

Fusion — no additional frameworks

Unlike wrapping, Fusion does not require any additional frameworks or libraries. As such, there are no conflicts with other frameworks, manipulation of load ordering or complexities related to parallel loading.

Appdome's Fusion™

Appdome's patented* Fusion technology is a general purpose, open integration layer for mobile applications. Fusion takes an SDK or any functionality and transforms it into something that goes beyond single purpose, single platform and single use case integrations.

With Fusion, multiple mobile services can be combined with a single app, quickly and painlessly. To use Fusion, a mobile developer needs zero access to source code. There are no additional agents loaded on the mobile device. All the capabilities are integrated directly into the native app binary without impacting application level APIs. Fused in services can co-exist inside the same application and provide better feature set granularity and choice.

With Fusion, the user experience, performance and functionality of the app are maintained. Even run time decisions can be made, enabling or disabling certain capabilities based on organizational policy so that unneeded features aren't running and thus aren't eating up system resources.

Wrapping vs. Fusion

Consider an analogy where a mobile device is a race car. Your mobile application is the engine. Inside the engine, your app does many awesome, very complex things that wow your users. Wrapping puts an envelop around the entire engine just to interact with the fuel management system inside your app. Doing so, wrapping intercepts, controls and modifies the app just to control a very specific function. It's overkill and often ends up slowing your engine down or worse, interfering with other functions that make your app so awesome in the first place. Oh, and did we mention, that wrappers only work with specific functions your app needs? There are no general purpose wrappers, not that you'd want one anyway.

By contrast, Fusion doesn't interfere with the entire engine. Fusion adds a Runtime Integration Module, that interacts, and connects, only with the function you intend it to. That way, the fuel management system can be connected to any other service or system without effecting the stability, performance and usability of your app. And better yet, Fusion isn't limited to just your fuel management system! Any service you intend to connect to your app can now be integrated simply and easily, opening the possibilities for expanded use and enjoyment of your app.

Appdome Automates the Integration Effort

Implementing Appdome is simple and quick. Simply bring the app binary onto the Appdome platform. The entire process of fusing capabilities like an EMM's SDK or mobile security feature set is automated. Leveraging an intuitive cloud-based web interface that can render a fused app within just a few minutes.

Try Appdome

The Appdome Platform offers an intuitive, fast and cloud-based service with minimal steps required to achieve the fusion of commercial and custom applications with a wide variety of third party functionality ranging from management SDKs to security capabilities and more.

Appdome's Fusion works as part of the application. So, your application can be signed using the same AppID as the underlying app. Even though the process is measured in minutes, the patents, engineering and expertise behind the scenes providing powerful, rapid and smooth workflow are extensive. Nobody else is addressing mobile integration this way.

Appdome provides broad platform support across all mobile development platforms, frameworks and operating systems like iOS and Android with full enterprise store and consumer facing store support via the AppStore and Google Play. Imagine, fusing target functionality in minutes without impacting or interfering with thousands of application level APIs or impacting your app's native functionality.

Appdome Extensibility

Appdome allows multiple SDKs, even from different vendors, to be fused to a single app. Operating with multiple SDKs has long been a wish that wrapping never granted.

Appdome also works seamlessly with IoT solutions found in retail, financial services, healthcare and critical infrastructure. This type of extensibility can only be achieved with Appdome as other approaches simply can't apply broadly to the varied number of IoT device types.

Since Appdome doesn't require access to source code and in fact works directly with binaries, the often troublesome process of trying to obtain app source code is eliminated. Now you don't need to limit integration based on source code availability.

Internally developed applications and commercial applications can be fused with management, security and other functionality within minutes instead of months. Application developers are frequently held to time-to-market metrics. With Appdome, third party integration time is virtually eliminated.

Appdome for ISV

Mobile application developers and independent software vendors (ISVs) work diligently to create great apps. They cringe at the thought of completing an app only to have it degraded by third party integration that sucks. When serving enterprise customers, mobile developers and ISVs need choice in how to bring their apps to market, meet the demands of the mobile enterprise, and maintain a quality experience for enterprise users.

A really nice feature of the Appdome platform is that it's open and designed to be intuitive and usable for anyone building, managing or deploying mobile apps. This also means that Appdome allows ISVs to reach more users and use cases with their apps, without depending on internal engineering resources to develop one-off solutions to integrate their mobile apps. If you ask us, that's pretty nifty.

Summary

Mobile application integration is not a new issue, nor is Appdome the first to offer a solution. Direct source code integration, SDKs, and wrapping are all approaches that developers have historically relied on to modify and import other functionalities into their apps. However, while these methods are not entirely without their individual merit, implementing them has proven to be either complex and time-consuming, or limited and unreliable — Appdome, by contrast, offers a new solution that overcomes the challenges with previous approaches and automates the integration process making the delivery enhanced mobile apps as simple as click, click, click.

Power in Simplicity

If you'd like to get started with Appdome, the steps are simple:

1. Upload an iOS .ipa or Android .apk package for fusing within the Fusion Cloud Platform
2. Select the desired functionalities to fuse to the app
3. Catch a few Pokémon for the next few minutes while the fusion occurs
4. Sign, download, distribute and install the newly fused app
5. Think about what you want to do with all that extra time

See? Appdome is mobile integration done right!

About Appdome

Appdome is a productivity platform for mobile integration, providing the rapid integration of multiple third-party functions to apps, shortening the deployment cycle and connecting mobile apps to other services on demand. The codeless service operates as a mobile integration workflow in the cloud, and allows users to perform integration projects on the final application package. No source code or development expertise is required. Likewise, no modifications to an app or an SDK are required to complete integration projects on the platform. The solution is currently used by the world's leading financial, healthcare and e-commerce companies to support productivity, compliance and security for consumers and employees. The company is based in Silicon Valley, United States and Tel Aviv, Israel. For more information, visit www.appdome.com.

*Yehuda et al. Method and a system for merging several binary executables. U.S. Patent 9,934,017 B2 filed November 15, 2015, and issued April 3, 2018.

San Francisco | Silicon Valley | Tel Aviv | +1.844.360.FUSE