Solving Mobile Fragmentation with Platform Agnostic Ecosystems
In today’s mega-fragmented mobile market, device agnostic development and distribution represents the future of mobile gaming, empowering developers to get their content ready for the cross-platform mobile market.

Contents
Introduction .................................................................................................................................................. 3
The Fragmentation Challenge .................................................................................................................. 4
Mobile Device Fragmentation .................................................................................................................. 4
  Real Life Example – Angry Birds ......................................................................................................... 5
  Additional Development Challenges ..................................................................................................... 6
Mobile Market Distribution Fragmentation .............................................................................................. 8
  Additional Mobile Market Fragmentation Challenges ............................................................................. 9
What is a Platform Agnostic Ecosystem? ............................................................................................. 11
What is Platform Agnostic Development? ............................................................................................. 11
What is Platform Agnostic Distribution? .............................................................................................. 11
  What are MoMinis’ Platform Agnostic Development Tools? ................................................................. 11
How Does MoMinis Do It? ................................................................................................................... 11
  What is Not Platform Agnostic Development? ..................................................................................... 12
What Are the Benefits of Platform Agnostic Ecosystems? ................................................................. Error! Bookmark not defined.
Conclusion ............................................................................................................................................... 14
Introduction

There's no shortage of information about the mobile fragmentation problem occurring in today’s mobile market. Just plug the keyword into Google, and you’ll find pages of tips, discussions, blog posts and analyst reports about the current and future implications of this subject for mobile developers, manufacturers and end-users.

Historically, mobile fragmentation addressed technological complexities. Discrepancies among devices and operating systems (OSes) made it exceedingly hard for developers to create hundreds of versions of games in order to reach a worldwide market. While mobile device fragmentation is still growing, fragmentation is also entering the realm of distribution. The competition for control over content consumption leads to a multitude of distribution options for developers.

The goal of this paper is to provide information about platform agnostic development and distribution ecosystems and how they allow developers to bypass pervasive fragmentation problems in today’s market.

A platform agnostic ecosystem refers to a unifying methodology of producing mobile content that is adaptable to all mobile device variations while attaining instant entrance to all distribution channels, using a method of ‘develop once deploy anywhere’.
The Fragmentation Challenge

While opportunities for game developers are on the rise as the mobile games industry booms and new devices are penetrating the market, developers are also facing ever-growing challenges resulting from this growth. The constant entry of new OSes, device specifications, upgrades, as well as application marketplaces creates complex device and distribution fragmentation that developers need to be equipped to overcome.

Mobile Device Fragmentation

Game developers and mobile entrepreneurs are more than aware of the challenges they face in creating mobile content amid worsening fragmentation in the mobile device market. The countless differences within platforms, OSes, device versions, screen resolutions, keypad types, and many other specifications are near impossible to comply with. Developers are confronted with enormous challenges as they try to create numerous modified versions of their applications to fit all device models in the mobile industry. Developers also face tough hurdles surrounding platform choice, expertise constraints, cost, manpower and other resource allocations in order to adapt to the shifting arena.

As new devices are flooding the market, fragmentation worsens and creates a vicious cycle for developers wherein their choices can also obstruct their success. In the lifecycle of a mobile game, the initial decision developers have to make lies in which platform they should target and is usually based on monetization and distribution potential rather than on the technical requirements involved. Developers will also usually choose one platform instead of multiple because of time and resource restrictions.

There are strong implications in making this decision at the beginning of the development process. Once a platform is chosen, developers are bound to it and limited to its capabilities as well as hindered from making educated decisions about other platform’s real market value. Once they choose a platform and operating system (OS), this in turn can cause a chain reaction that confines developers to its protocols, directly affecting the success of the game and the developers ROI.

"There is an alphabet soup of protocols, standards, and regional differences by country which can be daunting for any entrepreneur." - Richard Wong - Partner at Accel Partners

1 Rich_Wong - Partner @Accel_Partners - http://techcrunch.com/2010/03/04/mobile-fragmentation-forever/
Real Life Example – Angry Birds

In this multi–platform world, there are several poignant cases demonstrating how device fragmentation has caused critical performance problems in leading mobile games.

Angry Birds

In November of 2010, the fragmentation issue with Android phones was prominently highlighted when the popular mobile game, Angry Birds, encountered substantial malfunctions on high-end Android devices.

"With our latest update, we worked hard to bring Angry Birds to even more Android devices. Despite our efforts, we were unsuccessful in delivering optimal performance. So far, we have hesitated to create multiple versions of Angry Birds for the Android platform. We are currently developing a lighter solution to run Angry Birds on lower-end Android devices. This does not mean lighter game-play or a lesser amount of levels, but a game experience optimized for devices with less processing power," Rovio said.

2 Rovio -
http://www.rovio.com/index.php?mact=Blogs,cntnt01,showentry,0&cntnt01entryid=47&cntnt01returnid=58
Additional Development Challenges
The different development challenges that developers face are:

Platform Selection
The first decision that mobile game developers make is which mobile platform and OS they will build their games for (such as Symbian, BlackBerry, iPhone, J2ME, Android, Windows Mobile or others). Should they attempt to become an expert in one OS or try to tackle all of them, including all their sub-variant OSes?

Device Hardware Decisions
Even after selecting a platform, developers are still required to create numerous versions of games that can support different device hardware specifications within each phone. They should decide among supporting multi touch or single touch keypads, pointing devices, track balls, QWERTY keyboards, various keypads and even joystick controls. They are sometimes even required to make game editions that adapt to various screen resolutions from the highest end devices with HD screens (480x800px) to the lowest-end devices with low screen resolutions (128x128px). Additional parameters that should be taken into account are the processing power, hard disc space and memory capacities of the individual phones.

Learning Curve and Time Constraints
Becoming fluent in each individual platform’s programming language is a process that usually takes many months to years. Research has shown that Symbian, which currently has the largest global device market share, takes 15 months or more to learn while Android, which is predicted to surpass Symbian in market share, takes a reported average of 5 months or more to learn.

Expertise Constraints
Completing a high quality game and targeting single or multiple platforms is an operation that requires skilled programmers or teams of programmers. There are major barriers of entry into the mobile sphere for developers having less mobile expertise such as flash developers and regular PC game developers.

Maintenance
As upgrades and new versions of devices and OSes emerge, developers need to make sure their applications are compatible with all new and old versions of handsets in the market and update their code accordingly. This upkeep requires intense time allocation, expertise and attentiveness to market shifts.

Risk Management
Many risks are involved in embarking on a mobile game development project including making a bad platform choice, having slow or no market penetration, and receiving low or no return on investment.
Lack of information

Although we are flooded with information about the mobile market, it is sometime hard to find the truth behind the numbers. The idea of getting a game to market may sound relatively straightforward as one might assume that platforms with the largest amount of installed devices in the market should have the highest amount of apps available and consumers ready to download and purchase applications. However, upon further inspection, there is a major disparity between the number of devices being installed and the number of apps in each platform’s app stores. As seen in the right side chart, the iPhone has the second to lowest installed base of all devices though it has the highest number of apps available in their app store.

However, this doesn’t provide information about end-user consumption or the ratio between the numbers of applications per platform to the number of application that succeed to generate revenue. These numbers are sometimes hard to find, and moreover, change over time.
Mobile Market Distribution Fragmentation

The sheer number of choices to consider when distributing mobile content and the lack of centralized ways to easily access all channels leads to a very fragmented distribution market. There is an overabundance of native app-stores, multi-platform app-stores and other distribution avenues containing varying polices, revenue models, pricing points, and restrictions.

Some of the most popular distribution channels such as Android Marketplace, Apple App Store and GetJar each have varying business models with stringent rules and regulations. Different versions of games that fit the specifications of each app store might need to be created before developers can submit their games to each store.

Real Life Example – Android Marketplaces

The most glaring example of the fragmented mobile market distribution problem is evident in the great number of app-stores for Android content that have been declared since 2008 by Vodafone, Verizon, Telefonica and Amazon, in addition to Google's Android Market.

‘As of late 2010, there are 12 major Android app marketplaces and many less dominant players in this space. Google Android Market is the lead player. This creates a problem for Android developers. They cannot afford not to list their products on all marketplaces, yet it is simply too time-consuming to deal with different rules, processes, and store particularities, not to mention some markets are simply not accessible to a lot of developers. This problem is known in the industry as the “fragmentation of app distribution”’³. - Hubpages.com

³ http://hubpages.com/hub/android-market-publishing
Additional Mobile Market Fragmentation Challenges
The different distribution considerations and challenges that developers face are:

The Path of Least Resistance vs. the Multi-Faceted Distribution Approach

Some might think that the best game distribution approach is to take the path of least resistance by entering the most popular app stores in the market such as the Apple AppStore in the hopes of reaching the biggest audiences and highest revenue potential. This approach can at times be an excellent solution, however, there is incredibly more potential to be found in the many other accessible marketplaces. In addition, the Apple AppStore and other mainstream native Appstore are so saturated, that the chances to stand out and produce a hit through these channels is slim to none.

The other way to go for developers is distributing a game everywhere in a multi-faceted distribution approach. This approach, while theoretically covering all bases, takes incredible time and resource allocation. There are overwhelming amounts of distribution methods such as native app-stores, multi-platform app-stores, operator portals, direct download methods, OEM preloading, affiliate marketing and other marketing approaches. Some of these options are not even accessible to developers themselves.

In addition, submitting games to each of these channels means that focused attention needs to be given to each and every regulation and business model that apply to each store.

Market Penetration

Developers should also be aware that the number of apps available in an app-store does not correlate directly to the number of downloads made in this store. Additionally, leading app-stores such as the Apple, Android and Ovi store are so saturated, that it is extremely rare to stand out and gain exposure exclusively through these channels. A great marketing strategy is required to attain significant distribution. For example, adding a game to the lesser known app-stores in the market can be a great approach to reaching long tail niche audiences, however, it also requires a lot of work and manpower.

Monetization

Business Model Choice

One of the most important decisions is selecting a business model that fits the nature of a game. Is it a pay per download, subscription, ad supported, in-game purchasing or virtual currency model? One of the problems that arise is that different distribution channels allow differing business models to be used. For example, open platforms like Android allow the developer to implement any desirable business model, including subscription, rental and game packages. However, in closed environments like the Apple App Store, the developer is required to integrate proprietary billing systems (e.g. Apple’s billing SDK) that might not support any of the above.

Pricing

Figuring out the right pricing model usually requires lengthy periods of trial and error testing and plenty of time to identify the best models for each game per distribution channel. Developers should also stay attuned to how pricing varies depending on which area of the world they are catering to. Typically, countries with stronger economies will have higher pricing points.
**Developer Rev-Share**
Different distribution channels offer various rev-share models. When choosing your distribution channel, this number should definitely be taken under account. Astute developers will attempt to use diverse marketplaces in order to find which ones offer them the best ROI.

**Geographic Marketing**
One might think that choosing one dominating OS would easily lead to international consumer access. Nonetheless, various leading OSes control various regional markets of the world and developers must understand which devices dominate individual localized markets and tailor their marketing efforts in line with these regional trends.

**Content Regulations**
Every apps store and distribution channel has its individual content regulations which developers must adhere to and possibly adjust each and every one of their games to. A nice anecdote is Apple App Store’s regulation that prohibits games from having an exit button. That means that developers must specifically disable this functionality in their game if they want to enter the Apple market. Another poignant example occurred in January of 2011 when Kongrate, a games retailer, added one of their apps to the Android Market but was removed for what Google viewed as a violation of their non-competition terms of service. These cases illustrate how technically and legally diverse regulations in app-stores can interfere with the developer’s process of game distribution.

**Content Localization**
A game that is a hit in one country may have the potential to become an international sensation, but unless it is tailored to different languages and cultural mobile trends, it will not achieve those goals. Developers need to make sure their games are translated and localized to social patterns of mobile game consumption from various regions of the world.

**Time Constraints**
With all the responsibilities that come along with adapting to the fragmented mobile distribution market, the main stumbling block to success often boils down to time. Developers need to dedicate a great deal of man hours in order cover all the bases of distribution regarding picking marketplaces, choosing business models, building viable marketing strategies, upholding regulations and much more.
What is a Platform Agnostic Ecosystem?
A platform agnostic environment refers to a comprehensive method to develop and distribute mobile games that is unattributed to a certain OS or platform but is operational on any device, platform or distribution channel in the market. The Ecosystem is a single solution for overcoming both technological and distribution fragmentation by providing scalable means to reach any device and distribution channel automatically.

What is Platform Agnostic Development?
Platform agnostic development provides a solution to mobile device fragmentation. It refers to the method of developing mobile content once in an environment that is indifferent to the target platform that the game will run on. Using this technique, the developer develops for a virtual device while the platform itself is responsible for making games run seamlessly on different devices and OSs.

What is Platform Agnostic Distribution?
Platform agnostic distribution provides a solution to mobile market distribution fragmentation. It refers to the method of mobile content distribution that automatically covers various channels and is adapted to all business models, pricing points, regional trends and other regulations associated with game distribution. Using this technique, the platform itself is responsible for distribution with little to no effort on the part of the developer.

What are MoMinis’ Platform Agnostic Development Tools?
The notion of a unifying technology that will alleviate developer’s pains and open the doors to global mobile markets is something that the development world has speculated upon for some time but hasn’t been viewed as a realistic expectation, yet.

MoMinis, a company providing robust mobile game development and distribution tools, offers device agnostic development tools for developers to create, publish and receive ROI on their work. The MoMinis Studio is a RAD (rapid application development) tool that alleviates fragmentation by being agnostic to any specific platform or device. MoMinis’ revolutionary porting capabilities compile these games to be operational on thousands of handsets, including feature phones and Smartphones, within leading platforms such as Android, Blackberry, J2ME and Symbian.

MoMinis’ built-in distribution channels and partnerships with operators, online stores and publishers enable games to reach a global consumer audience.
What Is Not Platform Agnostic Development?

Virtual Machines
The Virtual Machines (VM) concept refers to a development environment layer that simulates native operating systems, thus allowing integration with them and enabling the "Write Once, Run Anywhere" runtime environment. However, the implementation of a single standard of a virtual machine is still open to interpretation and design. Different vendors use different approaches to implementing virtual machines on device’s OS. Add to that the different device versions, specifications, requirements and standards and you end up with excessive VM versions in the marketplace, each running with different APIs, bugs and OSs.

The Lowest Common Denominator Approach

One approach of solving fragmentation when developing for a single OS is to create a single light version that most devices will be able to handle. The problem with this approach is that even if developers are able to assure that their games runs smoothly on all various devices, a process that usually requires checking the game manually on each device, the developer will still need to make many compromises. A single version of the game will look very poor on high-end devices, or run very slowly on the low-end ones. This is not a problem that is going to go away soon as there are still new devices coming into the market that show poor performance. For example, the Sony Ericsson’s Xperia Mini is very limited in its capabilities as compared to Samsung’ Galaxy S.

One IDE – Different Kits

Some Integrated Development Environments (IDEs) enable developers to develop for various platforms by providing them with a set of SDKs (Software Development Kits). For instance, iOS and Android SDKs are provided by several widely used IDEs in the market. This solution creates a single version for each OS. However, this approach still requires developers to have different application versions for different OSs, which takes more resources and time to create and maintain. This approach doesn’t address the problem that different devices within the same OS usually have many major differences including varying processing power, input methods and memory capacity to name a few. A single version will probably not be compliant with all device specifications.
What Are the Benefits of Platform Agnostic Ecosystems?

Platform Selection
The essence of platform agnostic ecosystems lies in that the developer does not develop for one single platform, make any platform selection or have to consider the commercial and technical ramifications of a single platform. Rather, the developer writes one prototype game and has it supported by all platforms while entering multiple marketplaces simultaneously.

Learning Curve, Expertise and Time Constraints
Most mainstream platforms take months or even years to learn. Instead of spending years on gaining expertise for multiple platforms, a platform agnostic environment requires that the developer become fluent in a single programming environment, requiring expertise in only one IDE and lowering the learning curve radically. In addition, the platform agnostic ecosystem accesses all distribution channels, drastically shortening the time it takes to learn about and enter each channel manually.

Risk Management
When creating a prototype game in a platform agnostic ecosystem, it will operate on all platforms and the risk of choosing a losing platform is eliminated. The risk of choosing the wrong distribution outlet, revenue model, pricing type and marketing strategy is also removed as the ecosystem makes use of the widest variety of distribution methods.

Monetization
The ecosystem automatically adjusts business models for each designated platform marketplace. For instance, it can apply subscription based payments on android, freemium versions, premium payments on iPhones and pay per game models from operators. The ecosystem also automatically adjusts pricing models to fit marketplaces, game types and regional standards.

Market Penetration
Developing a game in an agnostic ecosystem escalates market penetration and monetization by enabling wide-reaching device coverage, allowing optimized access to any distribution channel.

Distribution
Using a platform agnostic ecosystem, mobile content becomes ready to publish on all platforms, opening many more avenues for distribution. Since the ecosystem utilizes wide-reaching distribution channels and revenue models, the developer’s potential for monetization and positive ROI is heightened.
Conclusion

Whether you are an expert mobile game developer with years of experience or a novice in the industry, the issue of fragmentation within devices and distribution channels is and will remain a huge hurdle to cross. This guide has outlined how mobile agnostic ecosystems lead to instant cross-platform device support, accesses major distribution channels automatically, and saves immense time and resources for developers in today’s mobile market. Agnostic game development is changing the face of mobile game development and is the most efficient and the practical choice for mobile game developers today. As the mobile gaming industry is booming, now is the prime time for developers to utilize the power of platform agnostic ecosystems and reap the maximum rewards from their mobile game ventures.

MoMinis Platform Agnostic Ecosystem

MoMinis Ltd is a company providing a robust mobile agnostic ecosystem for mobile game development and global distribution. MoMinis offers free development tools as well as comprehensive game distribution and monetization channels with a propensity for localization of games and geographical trends. For more information about MoMinis, please visit: [http://dev.mominis.com/](http://dev.mominis.com/)