

## ANDROID FRAGMENTATION VISUALIZED (JULY 2013)

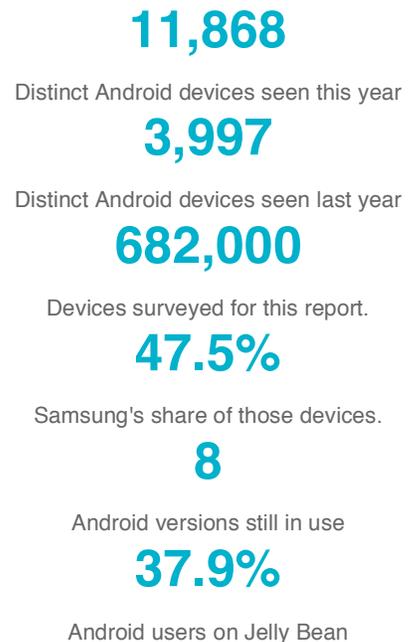
Fragmentation is both a strength and weakness of the Android ecosystem. When comparisons are made between Android and iOS the issue of different API levels, and the vastly different devices running them, is often emphasised. In this report we examine the extent of Android fragmentation and analyse its impact on both users and developers.

### The Problem:

Android devices come in all shapes and sizes, with vastly different performance levels and screen sizes. Furthermore, there are many different versions of Android that are concurrently active at any one time, adding another level of fragmentation. What this means is that developing apps that work across the whole range of Android devices can be extremely challenging and time-consuming.

### The Advantage:

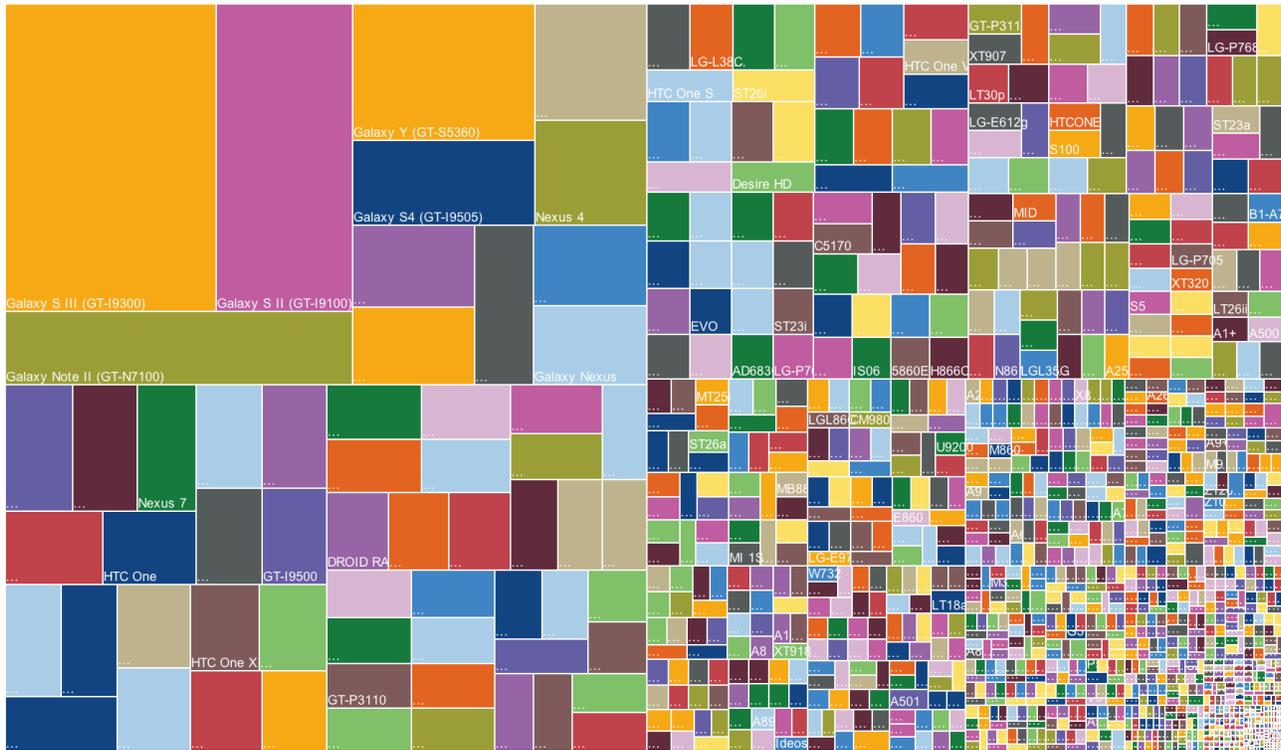
Despite the problems, fragmentation also has a great number of benefits – for both developers and users. The availability of cheap Android phones (rarely running the most recent version) means that they have a much greater global reach than iOS, so app developers have a wider audience to build for. It may be tricky to do, but the potential reward definitely makes it worthwhile. For consumers, extreme fragmentation means that they can get exactly the phone they want – big or small, cheap or expensive, with any number of different feature combinations.



## DEVICE FRAGMENTATION

This is the best way of visualizing the sheer number of different Android devices that have downloaded the OpenSignal app in the past few months. From a developer's perspective, comparing fragmentation from this year to the previous year, we see that it has tripled, with even more [obscure devices](#) from around the world downloading the app. If you want to understand the challenge of building an app that will work on all devices that want to download it, this image is a good place to start!

## Android Fragmentation 2013

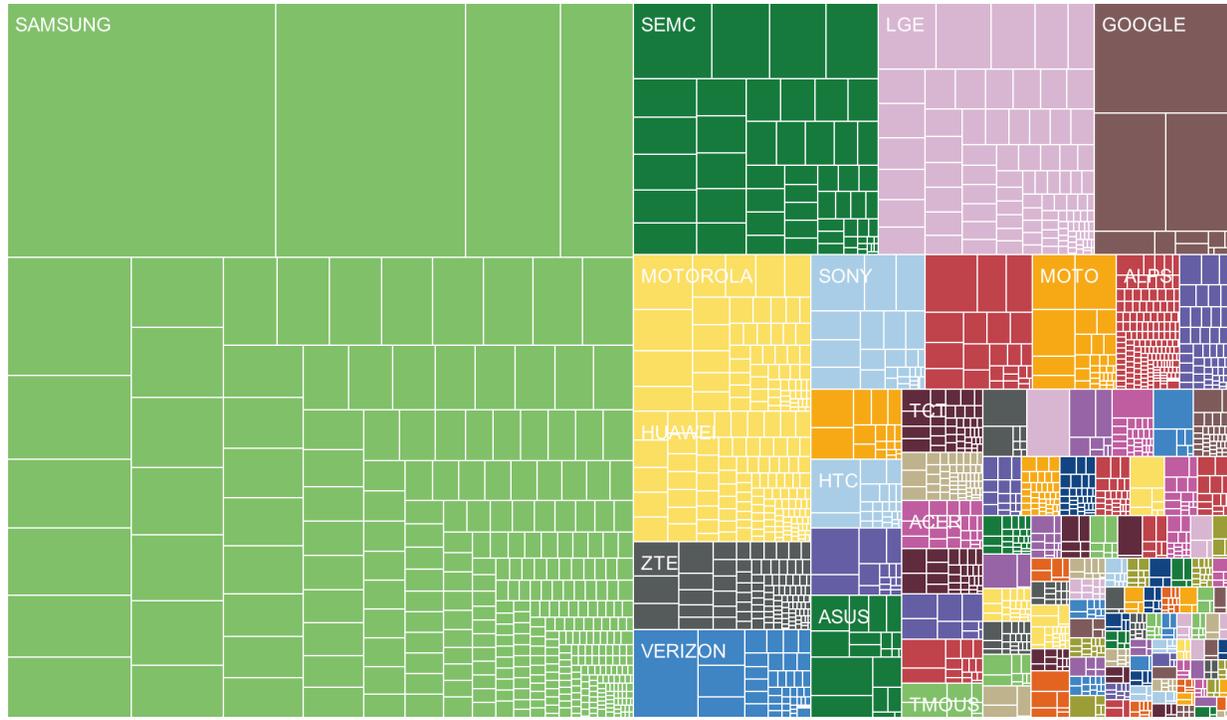


*We have seen 11,868 distinct devices download our app in the past few months. In our report last year we saw 3,997.*

## BRAND FRAGMENTATION

A similar look by brand, to see how much of the market each leading device manufacturer currently has, with Samsung clearly way out in front. Calculating the percentage share of the market held by the top few device manufacturers from our graphic really succeeds in emphasizing quite how dominant Samsung are, with Sony-Ericsson in second with a 6.5% market share - less than 1/6th of Samsung's. Some of the brand names shown as different in the graphic are part of the same company, i.e. Moto and Motorola are the same and HTC is shown as split up into its different regional variants. But even when unified under one umbrella name Motorola only ends up with a 4.2% share and HTC even less at 3.9%.

*Samsung have a 47.5% share of the Android market.*

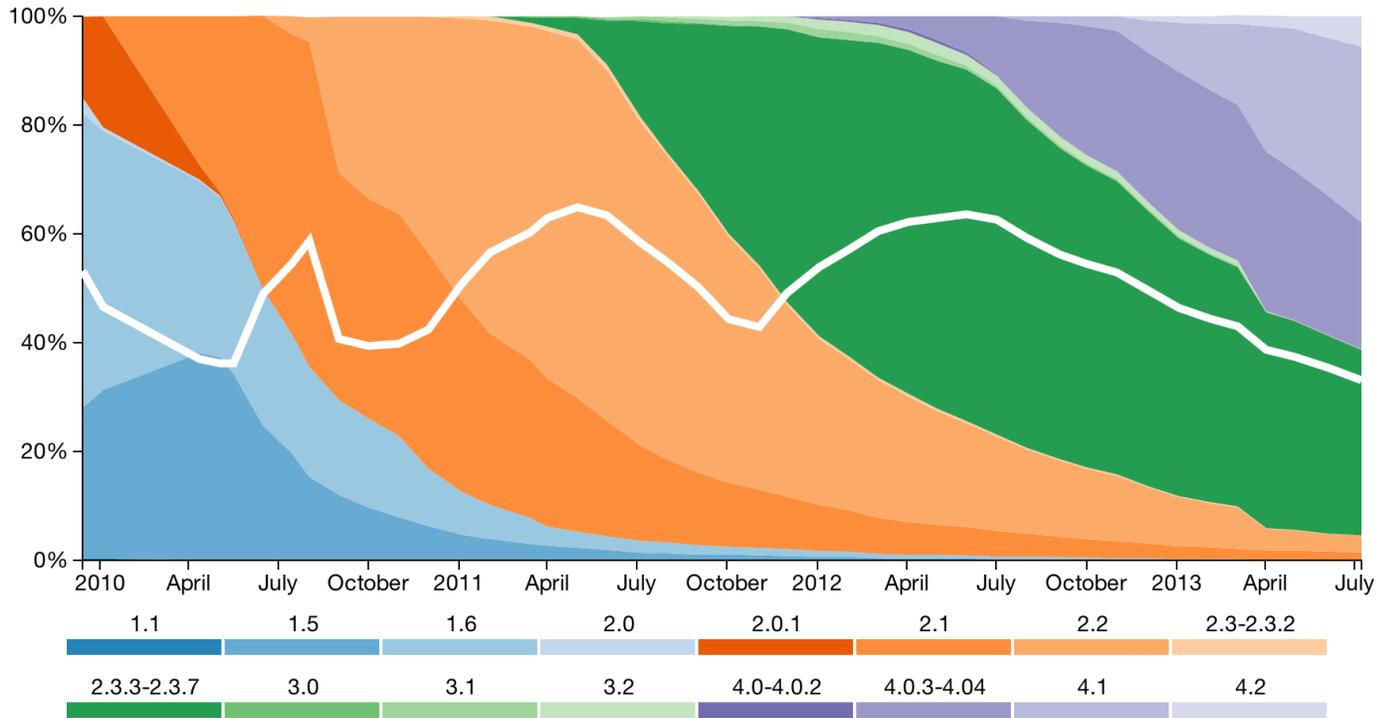


## ANDROID OPERATING SYSTEM FRAGMENTATION

Device fragmentation is not the only challenge that developers face when building for Android; the operating system itself is extremely fragmented and has only become more so over time. The above graph shows the relative stages of Android fragmentation, and the steady decline of any one Android version having prominence can be seen by the progress of the white line.

*The Android operating system is the most fragmented it has ever been.*

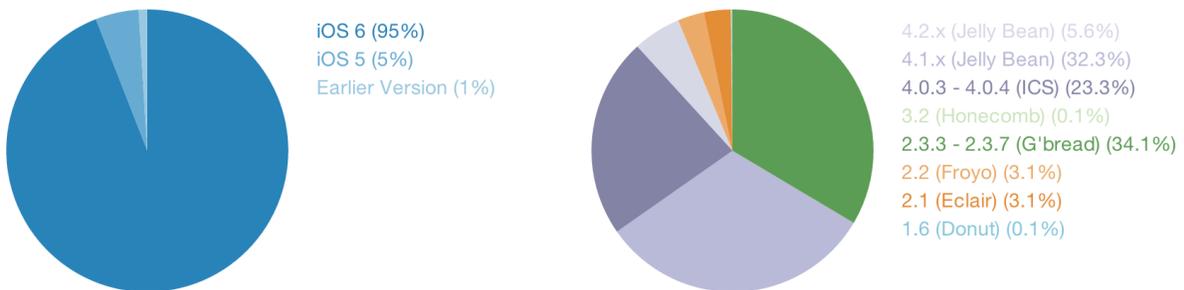
## Android Fragmentation 2013



The white line shows the market share of the leading API level at any time

## COMPARISON WITH IOS

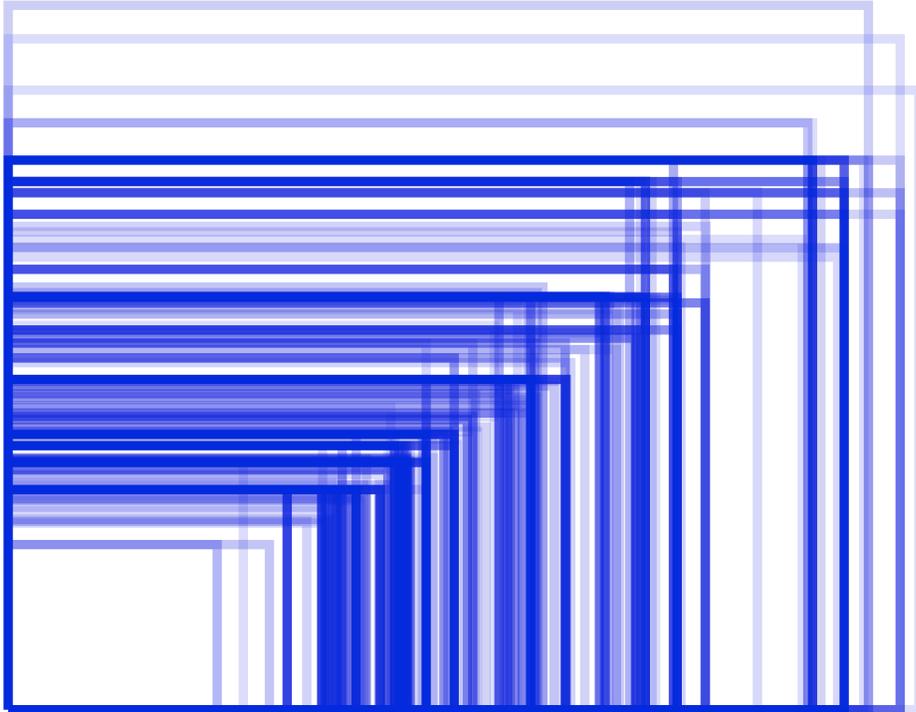
Android fragmentation of all kinds is usually illustrated in comparison with iOS. These two pie charts clearly show the difference in API fragmentation between the two competing operating systems.



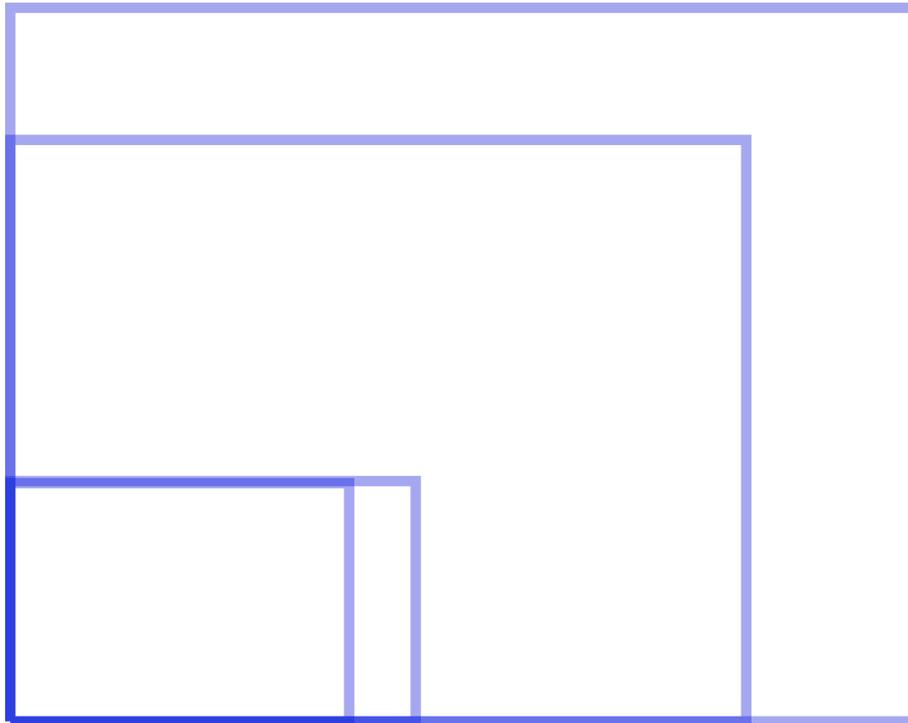
## SCREEN SIZES

Key to the success of any app is getting the UI right, and Android presents two particular challenges to developers in this regard. Firstly, brands have a tendency to produce their own variants on the system UI (Samsung's Touchwhizz and the HTC Sense being two such examples - which can change the look of

various default elements. Secondly, no other smartphone platform boasts such a proliferation of different screen sizes. For help in overcoming these difficulties see our [40 developer tips for Android Optimization](#).



Designing and coding layouts that work well across all these screens is hugely challenging. Across the dozen or so iPod-touch, iPhones and iPad varieties there are just 4 different physical screen sizes - partly due to Apple's tendency to double pixel density while quadrupling resolution (e.g. iPad 2 -> iPad 3) maintaining the same physical screen size. The graphic below shows iOS screen size fragmentation, allowing for an easy comparison with Android.



## SO WHAT DOES THIS MEAN?

Device fragmentation is increasing, and with it the reach and density of the Android operating system. While fragmentation certainly poses a headache to developers who have to test and optimize on an ever-increasing number of devices, the success of the of the Android ecosystem cannot be separated from its fragmented, free-for-all, nature. Similarly, it is easy for developers to bemoan API level fragmentation, but this is part-and-parcel of device fragmentation. Cheaper devices will struggle to run the most recent versions of Android and the fragmented operating system serves as an enabler of an ecosystem that is becoming more globally, and socio-economically, inclusive.

Fragmentation is responsible for a broader, more geographically dispersed, Android market - but fragmentation isn't simply limited to devices and operating systems. It is also important for developers to think about the impact of contextual fragmentation, the variety of differing contexts in which devices are actually used. What is relevant for one region may not be to another, and developers need to take into account differences in network performance and reliability when designing their apps - as well as the level of Wi-Fi access for apps which are particularly data heavy. Another relevant factor is battery life; while one day's battery life may be acceptable in the developed world, it may well not cut it in developing markets. It is important to remember that the criteria against which app performance is judged can change by region, not simply by device.

What is clear from this report is that Android fragmentation, of all varieties, is increasing. Too often this is treated as a problem with Android, rather than a strength, but we feel that this misses the bigger picture.

While there are certainly problems associated with fragmentation (and as developers we know them all too well), it is wrong to suggest that it is only a downside. Apple are currently working on a lower-end device, increasing the fragmentation of their ecosystem in the process, suggesting that the Android ecosystem is not only doing something right, but doing something to be imitated.

## NOTES ON METHODOLOGY

---

**The Device Fragmentation Graphic** shows the 11,828 distinct device types that were present in the last 682,000 unique devices to download our app. The reason we chose 682,000 is because we wanted to make a fair comparison with last year's fragmentation report, which was based on a sample of 682,000 devices over a set period of time.

**The Brand Fragmentation Graphic** is based on the same 682,000 devices as the device fragmentation graphic.

**The API Level Fragmentation Graphic** is based on data made publicly available by Google (though it involved some digging). The stats up until Feb 2012 were collated by Chris Sauve [here](#) and he has our eternal thanks. Also a thank you to /u/ikjadoon on Reddit who pointed us in his direction!

**The iOS Pie Chart** is based on information made publicly available by Apple.

**The Android Screen Size Graphic** is based on information from 3.41 million OpenSignal users. In this graphic we are showing physical screen size, not size in pixels. Changes in resolution at the same screen size (for example as between the iPad 3 and iPad 2) present fewer difficulties for developers than changes in physical size. We derive the screen size by dividing the number of pixels of height and width by the pixel density (points per inch - ppi). Ideally we would use separate values for X-density and Y-density as screens are sometimes manufactured with different ppi in different directions, however Android only gives access to one variable. Also worth noting: with rooted devices users are able to "alter" the pixel density - in reality the physical characteristics of the screen stay the same, but the Android OS thinks it's running on a larger device.