PLATFORM CHOICE BY MOBILE APPS DEVELOPERS*

Timothy Bresnahan, Joe Orsini and Pai-Ling Yin

Stanford University

February 13, 2014

For the past two years, Apple's iOS and Google's Android operating systems have split the market share of smartphone devices and the mobile applications (apps) for those devices. We model and estimate the platform choice by mobile app developers, including the decision to multihome. Our model flexibly models the potential gap between the decision to multihome and the realized demand from that decision. We find far less difference in preferences across platforms than across types of developers and apps. We identify strong incentives for developers of the most popular apps to multihome, making tipping unlikely.

*tbres@stanford.edu, jorsini@stanford.edu, pyin@stanford.edu. This research project is based on data collection and analysis over a wide range of data sources. We are very grateful to a number of research assistants who have worked on those datasets, gathered industry information, and joined us in industry interviews. These include Markus Baldauf, Sean Batir, Robert Burns, Jane Chen, Emanuele Colonnelli, Elizabeth Davis, Sherry Fu, Osama El-Gabalawy, Carlos Garay, Jorge Guzman, Alireza Forouzan Ebrahimi, Tim Jaconette, Nayaranta Jain, Julia Kho, Sigtryggur Kjarttansson, Xing Li, Derek Lief, Sean Mandell, Laura Miron, Jaron Moore, Yulia Muzyrya, Abhishek Nagaraj, Jin Hyung Park, Francis Plaza, Hatim Rahman, Juan Rios, Sam Seyfollahi, Melissa Sussman-Martinez, Masoud Tavazoei, Sylvan Tsai, Julis Vazquez, Sarah Wilson, Joon Young Yoon, Jessica Zhang and Parker Zhao. We are also very grateful to the many industry participants who have shared their time and expertise with us.

Platforms have often been used in information communications technology (ICT) industries to successfully harness innovations from diverse sources. The application development platform is the industrial organization used to create value out of information technology (IT). Applications developers convert a technical opportunity into economic value. A platform host will invest in the infrastructure, both technological and institutional, to coordinate application innovations for some general purpose technology. This lowers the cost of innovation immensely for the application developers. Especially for a new industry, platforms create an opportunity for disruptive innovations to emerge from the surge of entrepreneurial entry. The subsequent variety of applications developed attracts end-users, inducing further developer entry. The platform exploits social increasing returns to scale to produce diverse successful applications, increasing the economic value of the technical progress embodied in a platform's general purpose components.

The indirect network effects from this platform environment drive towards a tip to a single platform. Fragmentation of developers and consumers across multiple platforms is costly in several ways. The deployment of multiple platforms involves wasteful duplication. This duplication is true in any increasing returns industry, but in platform industries, the costs of duplication are borne in substantial part by users and developers rather than by platform suppliers. Both sides may delay entry or adoption due to uncertainty about when the market will tip and to which platform, diminishing the reinforcement power of the indirect network effects on any particular platform, and leading to slower diffusion of the underlying platform technology and application innovations across all platforms. Developers who enter fragmented markets face the cost of a smaller pool of demand if they choose one platform, since potential customers are spread across multiple platforms, or the cost of multihoming (developing for both platforms) in order to reach their entire potential demand.

However, when we examine the most recent and important example of an ICT platform industry, the mobile application (app) ecosystem, the industry seems to have stabilized as a market split evenly between the Apple iOS and Google Android platforms. This industry started in October, 2008, with the release of the first apps for the Apple iPhone on the iTunes store. Although existence of multiple platforms at the beginning of an industry is not uncommon, this industry has been stably split on both the end-user and developer side for the past two years. What explains the lack of tipping in this industry? Will it continue?

To answer these questions, we propose and estimate a model of developer choice of platforms (including the choice to multihome) in the mobile app industry. We use data on developers' platform choices and on the app's usage on the entered platforms to estimate the developers' expected profitability from entering either or both platforms. We have a new solution to the problem that the list of potential entrants may be selected. Our estimates suggest the following results salient to a potential platform tip:

1. Android and iOS are roughly equally attractive as platforms to US developers. This is consistent with the observation that neither platform has attracted significantly more applications than the others. The tie in attractiveness appears to hold across categories of apps, so that the existing spread of developers across platforms market is *not* due to observable sorting of differentiated developers to differentiated platforms, but instead reflects smaller, idiosyncratic errors in developer preferences. This suggests the developer side of the platform market is tippable, in the sense that there is no body of developers with strong incentives to stay with a minority platform.

2. Large developers and developers from established (outside mobile) firms are more likely to multihome. The expected success in terms of users for these developers is consistently high on both platforms, so it is worth it for these developers to incur the fixed costs of multihoming. Thus, even though a substantial number of apps do not multihome, from a user perspective the most popular apps are multihomed; an index-number calculation suggests that most of the contribution from app availability to the attractiveness of either platform to users comes from these large apps. This suggests that the user side of the platform market has little incentive to tip, and that a very large shock would be needed to lead most users to choose one platform over the other. A tip seems far in the future.

3. With no near term tip likely, platform fragmentation will likely persist. Platform fragmentation has different implications for different categories of market participants. Users are hardly impacted in the short run, and large developers spread the fixed costs of multihoming over large demand. Entrepreneurial developers, however, are strongly impacted. The idea that mobile platforms would lead to wide ranging entrepreneurial experimentation with market-wide implications is undercut by this impact.

Ultimately, we identify an asymmetry between large, established firms and entrepreneurial firms, but not in the way many anticipated: the costs of platform fragmentation are borne by the smaller entrepreneurs, while the larger firms are able to overcome these costs and multihome. In this setting, the potential for disruptive entrepreneurial innovation is diminished. Our discussions with industry participants suggest that these costs are marketing costs to become visible to the consumer; large, established firms avoid these costs due to their existing consumer relationships. Furthermore, the platform equivalence and multihoming by the most demanded apps suggest that the mobile apps industry is unlikely to tip.

The paper is structured as follows: Section I. reviews the relevant literature. Section II. describes the industry setting of mobile apps. Section III. describes our model of developer choice over platforms and multihoming. Section IV. describes our data. Section V. describes how we implement the economic model to analyze our data. We discuss our findings in Section VI.. The last section concludes.

I. Literature Review

Most of the economics literature on the mobile(wireless) industry studies wireless carriers like AT&T and Verizon. A new literature studies hardware devices, such as the Apple iPhone, in connection the carriers (Wilkinson (2013), Chintagunta et al. (2013)). Our closest antecedent is the work of Kevin Boudreau on applications development in the early handheld device era preceding smartphones (Boudreau, XXX). While Boudreau focused on the organization and management of the platform, our contribution, in this and in earlier papers (Davis et al. (2013), Bresnahan et al. (2013)) has been to look at the competition in the software that defines the platforms (iOS, Android, Windows mobile, etc.) and at the competition in application software that creates economic value on top of those competitive platforms.

A very successful literature studies the choice of platforms or, very similarly, standards, by suppliers of complementary products. Rysman (2004) is the classic paper, where the complements to a yellow pages "platform" are advertisements. Some of this literature has taken up the question of complementor choice in applications platforms (Lee (2013), Dube et al. (2010)) and Boudreau (XXXX) have considered changes in platform policies, such as openness. Most of these papers perforce study a mature platform or two sided market after the races, if any, for standardization have been run. Only a few papers (Augereau et al. (2006), Dranove and Gandal (2003), Brown and Morgan (2009), Cantillon and Yin (2013), Church and Gandal (1992), Corts and Lederman (2009)) study the race itself. Uniquely among these papers, we study a development platform (as opposed to a trading platform) in a context where multihoming on one side of the market is possible.

Although the theoretical literature on suggests a number of reasons why platforms may coexist (Economides and Siow, XXXX, Ellison & Fudenberg, 2003; XXXX, XXXX), only Brown & Morgan (2009) and Cantillon & Yin (2013) empirically analyze horizontal and vertical reasons for coexistence. Again, their work examines trading platforms, and their markets ultimately tip. We contribute to the literature by identifying market forces which would cause platform coexistence to persist, and we therefore find different reasons than those identified in Brown & Morgan (2009) and Cantillon & Yin (2013).

We model the developers' decision to write for a platform as entry into the market defined as users of that platform. As a result, our work builds upon the market entry literature, reviewed in Berry and Reiss (200x). The central inference of the entry literature arises from studying the choice of potential entrants to go in to or stay out of the market. A large literature, descended from the work of Berry (1992) and Sutton (1991), identifies the list of potential entrants into a market as the actual entrants, present or past, in another market or markets.[1] This obviously could create a problem of selection if profit in the market

1. Another approach avoids the problem of selection by specifying a list of niches that might be entered rather than the list

at hand is correlated with profit in the market(s) used to define the list of potential entrants. We follow in this tradition, and our model explicitly addresses the problem of selection in the list of potential markets.

We deal with the selection issue by writing a single model that jointly predicts entry into two markets. In each market, we observe not only the fact of entry but also, if demand is above a threshold, the number of users. We condition on satisfying the minimum threshold of quantity demanded on at least one platform, and construct a model in which we can deal with the selection of potential entrants into each platform. Our identification strategy requires parametric assumptions about the shape of returns to entry; because we observe demand as well as entry, these assumptions are testable.

Another advantage of this sample approach is that we are identifying multihoming costs off of developers who we know are capable of producing valuable apps. This allows us to address questions about the contribution to either single platform competitive advantage or to multihoming, i.e., to consumer advantage, without regard to consumer platform choice.

II. Industry Setting: Mobile Apps

An outstanding and dynamic example of the platform industrial organization is the dramatic rise in consumer use of mobile devices and the explosion of applications software running on those devices – mobile “apps." The invention of mobile app platforms has permitted developers to offer a system solution to their customers – re-using, not reinventing, the technology of mobile phones, mobile transmission, wi-fi, cloud technologies, and many other components.

The newness and popularity of the mobile apps industry has created many misconceptions about the industry structure and its practices, so we use this section to clarify the facts. Figures I and II show the history of users and apps on both platforms in the US. It is apparent from these graphs that the industry is currently in an even split of market share, and has been in this fragmented state for quite a while given then short length of the industry's entire history.

In theory, and as speculated in the industry press, there are a number of factors that may explain the fragmentation in the installed base. On the consumer side, the demographics of the iOS and Android users are in fact different: Android users tend to have lower incomes, for example. The split between Android and iOS may be a vertical differentiation over the cost of the device. It may also reflect differentiation in preferences over the device. These may also be correlated with differentiated preferences in apps, which would then lead to developers splitting themselves between the two platforms. It is possible that consumers are multihoming with multiple smartphones, but this behavior is rare enough to not matter at a market

of potential entrant. Examples include Bresnahan and Reiss (1991), Seim (xxxx) and Mazzeo (xxxx).

Figure I: Fragmentation of users. Source: http://www.tech-thoughts.net/2012/07/global-smartphone-market-share-trends.html#.UtdYR_RDtnJ

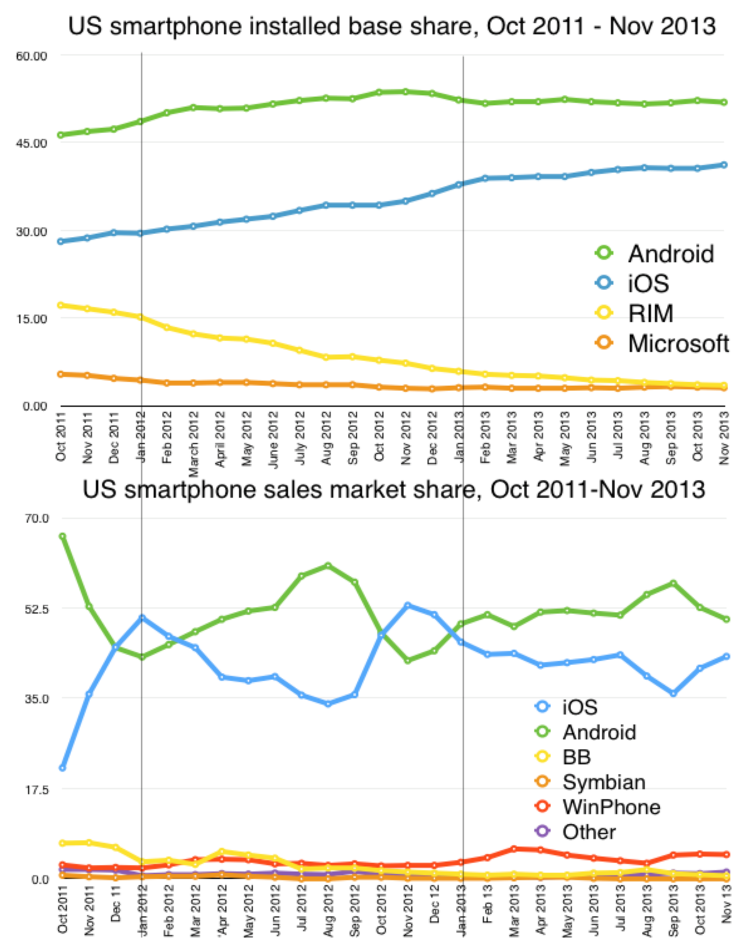
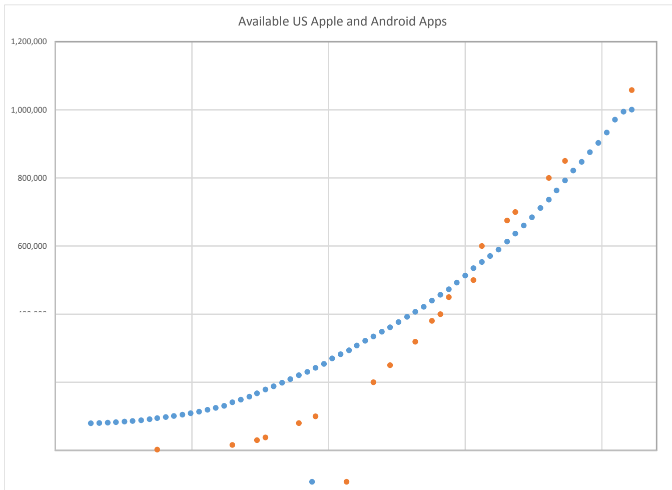


Figure II: Fragmentation of Developers. The solid dotted blue line is iOS apps. The sporadic orange dots track Android apps. Source: http://148apps.biz/app-store-metrics/?mpage=appcount, http://en.wikipedia.org/wiki/Google_Play#Applications, and http://www.appbrain.com/stats/number-of-android-apps, all accessed January 15, 2014



level.

Even if the preferences of consumers did not drive developers to different platforms, there are additional reasons that we might find an equal number of apps for both platforms. When choosing between two platforms, developer profitability depends not only on the relative size of the installed base of users, but also on the technical and contractual features. The technical features of an applications platform are a driver of application development cost, as the general-components in the platform are combined with the specific innovation of the application to create a usable product. Applications platforms often offer contracts, rules, and restrictions to developers that affect their profitability. There are a number of ways in which the iOS and Android platforms differ on these dimensions from the perspective of the developer (see Bresnahan et al. (2013)). Finally, developers might be multihoming, if the costs of multihoming are low enough and the benefits are large enough. The incentive to multihome is particularly strong when platforms are approximately equal sized. Indeed, if the benefits to reaching a large body of users on the other platform is high enough, it could outweigh even the benefits of platform differentiation that might drive a developer to favor one platform over the other.

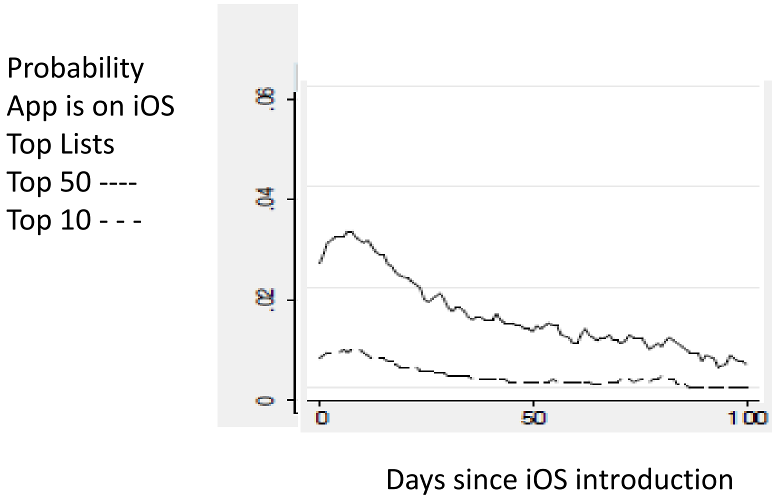
One very important fact to note from Figure II is the huge number of apps on both platforms. These huge number of applications may make it easier for the consumer to adopt the iOS or Android platform without too much consideration of which platform has the most applications: indirect network effects for the consumer may be overwhelmed by a critical mass of apps on each platform that provide sufficient utility. However, this same fact makes it extremely difficult for app developers to gain visibility and get matched to the right consumer. The developers may be spreading across platforms to reduce competition as much as possible.

One way in which the platforms do not differ significantly is the mechanism through which they predominantly match apps to users: top lists. These top lists recommend apps based on aggregate lagged user downloads, employing typical collaborative filter methods to try and identify the most useful or popular apps. The most important top lists cut across app categories, since the app categorization scheme is weak. Unfortunately, since the stores and these mechanisms are effectively the only channels through which apps can reach their customers, they create incentives for apps to game the system and try to "buy" their way into the top rankings. The commercialization / distribution facilities used by the platform providers reward success for app developers with more success, and reward near misses on success with very limited visibility. Thus app developers impose positive externalities on one another through the usual platform positive-feedback mechanisms, but also impose negative externalities on one another through congestion in visibility.

Even without these distortions, simply trying to get noticed out of the close to 1 million apps competing for a users attention will require marketing expenditure for most apps, and the success of this expenditure is not assured. Our many discussions with industry participants corroborates patterns we find in the data regarding the high costs of marketing an app. In Figure III, we profile the probability that apps make it into the Top 50 and Top 10 lists. The initial increase in the first few days after launch of the probability of being in these top lists is often fueled by apps who typically invest in marketing through "incentivized downloads" (purchasing downloads from users) and advertising their app in other apps for the express purpose of increasing their standings in the rankings. These launch campaigns average approximately $0.5 million. Despite these efforts, the probability of being on the top lists drops dramatically in the 2nd through 4th weeks, indicating that for many of the apps, these incentivized downloads and advertising campaigns did not work.

These industry characteristics suggest that two sets of costs face the developer, creating a gap between the decision to supply a given platform and the realized profitability on that platform. The first set of costs, the technical costs of porting, are determined by how feasible it is to write on the other platform. The second set of costs are determined by the market costs of being profitable on the other platform. Note that the second set of costs are defined by the market institutions implemented by the platform host. Each of these costs may be larger or smaller for developers of different types.

Figure III: Costs and Risks of Buying into the App Store Rankings



III. Economic Model

We model developer and user value from choosing a particular platform. The developer and users models differ both economically and econometrically. In this section, we lay out the economic models.

III.A. Developers

Developers choose on which of two platform(s) p to publish app a: $p = i$, the iOS platform; $p = d$, the An$d$roid platform; or $p = b$, both platforms. Throughout this section we suppress the subscript a. We model this as deciding which applications market(s) to enter, where the market boundaries are defined by the platforms. A developer who only publishes an iOS app can sell to iPhone users but not to Android users. Like all entry models, our model assumes that the developer has fixed costs and will only produce if the size of the market is sufficiently large to cover them.

We model variable profit for each app as a linear function of the number of customers, following entry models generally, and label the variable profits per customer on platform p as M_p. For platform economics, it is important to distinguish between U_p, the number of users of platform p, and r_p, the app's "reach" on platform p, measured as a the fraction of U_p that use the app. The number of customers for the app is $U_p \times r_p$, but the two parts have different economics. The overall attractiveness of the platform determines U_p, while the attractiveness of the particular app itself determines r_p.

Finally, let the fixed costs of publishing an app on one or more platforms be C_p , with $C_b \leq C_d + C_i$. We note that C includes both the technical costs of writing the app and the marketing costs of introducing it to customers. The variable profit of an app on a platform is $\pi_p = U_p \times M_p \times r_p$. It is profitable to write the app for platform p if

$$U_p \times M_p \times r_p \geq C_p. \tag{1}$$

It is profitable to write for both platforms if (1) holds for both i and d or (a weaker condition) if

$$U_i \times M_i \times r_i + U_d \times M_d \times r_d \geq C_b. \tag{2}$$

Finally, prior to entry, a developer may not fully know profits. We assume that the developer knows U_p and M_p and has a signal of r_p, which we call $\widetilde{r}_p$. Let $\widetilde{r} = (\widetilde{r}_i,\ \widetilde{r}_d)$ in an obvious notation. This lets us characterize the conditions determining $S = (S_i,\ S_d)$, the developer's choice of whether to supply a given app to each platform. Assume a risk neutral developer, and denote $\widetilde{\pi}_p = E[\pi_p|\widetilde{r}]$ as the developer's forecast of variable profits on platform p based on both reach signals (both will matter in general). Then the conditions for each supply choice are

$$\begin{aligned}
&S = (1,\ 1) \quad \text{if} \quad \widetilde{\pi}_i + \ \widetilde{\pi}_d \geq C_b \\
&\text{otherwise} \\
&S = (1,\ 0) \quad \text{if} \quad \widetilde{\pi}_i \geq C_i \ \&\ \widetilde{\pi}_i + \ \widetilde{\pi}_d \leq C_b \\
&S = (0,\ 1) \quad \text{if} \quad \widetilde{\pi}_d \geq C_d \ \&\ \widetilde{\pi}_i + \ \widetilde{\pi}_d \leq C_b \\
&\text{otherwise} \\
&S = (0,\ 0)
\end{aligned} \tag{3}$$

We have no model of pricing as a determinant of profit because price plays little role in developer profit for most apps. Corporate apps typically add to the profit of other products, as, for example, the United Airlines app contributes to the sales of tickets. Entrepreneurial apps typically attract an audience that can be monetized, either by in-app payments or by advertising. Since the price of the app itself so rarely plays a role, we do not model it and instead include a per-user profit in the model.

III.B. Users

Users choose a single platform in our industry, in contrast to developers who can and do multihome. From an indirect networks effects perspective, what matters is the aggregate demand for a platform by users, so we focus our user modeling on an index of the contribution of the apps available on a platform to the attractiveness of a platform to a large number of users.

Our model of users will not lead directly to a user choice model for estimation but instead to an index of the value of the all the apps available to consumers on a platform. Thus we now shift our focus from each app to all apps. The set of apps on a platform, N_p, and in an obvious notation we say that app a is available on a platform if $a \in N_p$ and that app a has reach r_{pa} on p. Since most apps are free, the usual index number problems for discrete choice are not present.[2] Reach is a measure of the quantity demanded of an app, in the specific sense of the fraction of users who choose to use the app. Accordingly, we measure the contribution of all apps to the total attractiveness of a platform as $\sum_{a \in N_p} r_{pa}$.

III.C. Three Simplifying Assumptions

First, we assume $C_b = C_i + C_d$. This is an economic assumption that the platforms are very different, i.e. that there is no fixed-cost savings from writing for both. In our context, we make this assumption because of what developers have told us about their cost function. It is not that porting is extremely difficult technically, which would tend to make the C_b close to $C_i + C_d$. Instead, it is the large size of platform-specific marketing costs.

Second, we change the units of the fixed costs and write $\kappa_p = C_p/M_p/U_p$. This is simply a normalization. We will estimate κ; the normalization determines the interpretation. In our application, U_i is slightly smaller than U_d, so if we find that $\kappa_d > \kappa_i$ the interpretation is either that the fixed costs of writing an Android app are higher $C_d > C_i$ or that the per-customer profits of an Android app are lower $M_d < M_i$. We cannot distinguish these two hypotheses.

Third, in our statistical model of app supply, we normalize κ_p to be a constant across apps of a similar observable type, so that variation in profitability across apps is driven only by r_p. This is an economic assumption; since $\kappa_p = C_p/M_p/U_p$, this is the assumption not only that there is no fixed-cost errors[3] but also that per-customer profits are constant across apps.[4]

We can now simplify the developer's supply choice in Equation (3) as

$$
\begin{aligned}
S_i = 1 &\Longleftrightarrow \quad \widetilde{\pi}_i \geq C_i \\
S_d = 1 &\Longleftrightarrow \quad \widetilde{\pi}_d \geq C_d.
\end{aligned}
$$

2. See Small & Rosen (1981).

3. The distinction between a random effect in fixed cost and a random effect in variable profits will not matter much for platform choice. See Bresnahan & Reiss (1991). Since we have a continuous-valued dependent variable, reach, we do need a variable-profit/market size error.

4. This assumption will be less problematic if we include regressors that capture the main observable variation in profits per customer in κ. These include indicators for in-app purchases, advertising, and some measure of high-engagement categories (e.g., the games category indicator).

IV. Data

We have gathered a sample of apps which were written for either iOS (iPhone), or Android phones, or both. We gathered our primary data set from public sources with the help of research assistants. We asked them to download each app in the sample and to fill in a questionnaire with 200 questions regarding the app's use of advertising, in-app purchases, and other monetization strategies. We also asked them to visit the developer's website to learn about the app's platform availability and developer characteristics.

We match these data to the January 2013 Mobile Metrix dataset from comScore, filling out our questionnaire for apps as necessary to ensure full coverage of the comScore apps. Two panels, one with Android phones and the other with iPhones, of approximately 5,000 US adult users, allow comScore to track their possession and usage of apps. We observe their data aggregated to the app*platform*month level. In those data, however, we only observe apps which meet a minimum usage test for each month on each platform: comScore includes data on the app only if it is used on that platform by more than 5 (at least 6) unique users.[5] As a result, while we may know from our primary data set that $S_p = 1$, if the app does not meet the usage criterion on a particular platform, then nothing about it is reported on that platform, i.e., $S_p^* = 0$, where S_p^* and the corresponding S^* denote the supply of an app on a platform as reported by comScore. We further drop apps produced by Apple, Google, carriers, and OEMs, which are typically pre-loaded onto devices, so that an observation is an app developed by an independent software vendor. This results in $1,044$ apps.

We observe S, S^*, and r^*, comScore's report of its estimate of the app's "reach" for each platform on which the app has met the usage criterion. Note that r_p^* is censored by the comScore sampling rule that it is not reported unless $S_p^* = 1$. This set of indicator variables concerning the developer's publication decisions and the app's satisfaction of comScore's usage criterion, together with the reach of the app when that criterion is met, form the dependent variable in our model.[6]

Figure IV shows a histogram of apps and their observed reach r^* (horizontal axis) in our sample. The vertical axis is the fraction of apps which fall into each of the equally spaced bins. This market is very skewed on both platforms, with most apps being in the lowest bin of reach (the far left bin), and a very few apps reaching almost all users (the far right bin).

Figure V shows the distribution of S and S^*. Each slice of the pie chart shows one value of (S, S^*). The first letter in each pair indicates whether the app was written for both platforms or only Android or the iPhone, based on data collected by our research assistants. The second letter in each pair indicates whether

5. comScore actually tracks apps with less than this level of usage if a client has requested tracking. We drop these apps to ensure uniform truncation of observations from the comScore dataset.

6. We multiply the "reach" estimate on each platform by 5,000, the size of comScore's platform-specific sub-panels, to back out the number of panel-members who used the app.

Figure IV: Distribution of Apps over Reach. Source: comScore, January 2013. This histogram of app reach has equally spaced bins. The vertical axis is the fraction of apps in the bin.

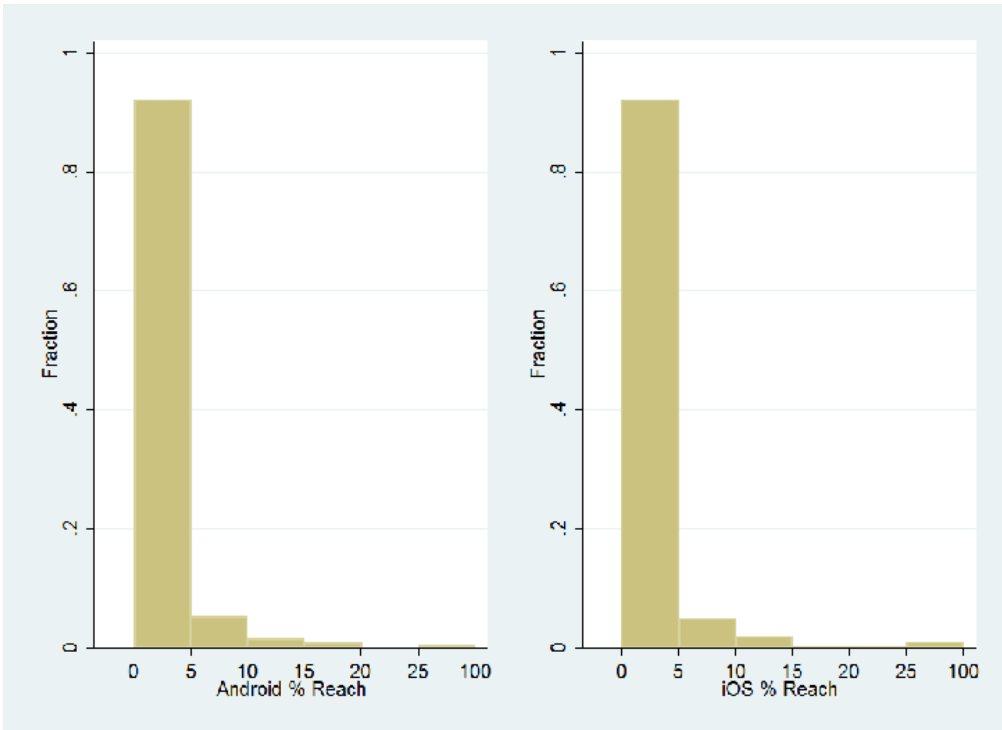


Figure V: Platform Choices (S, S^*) with b= both, d=Android, i=iOS.

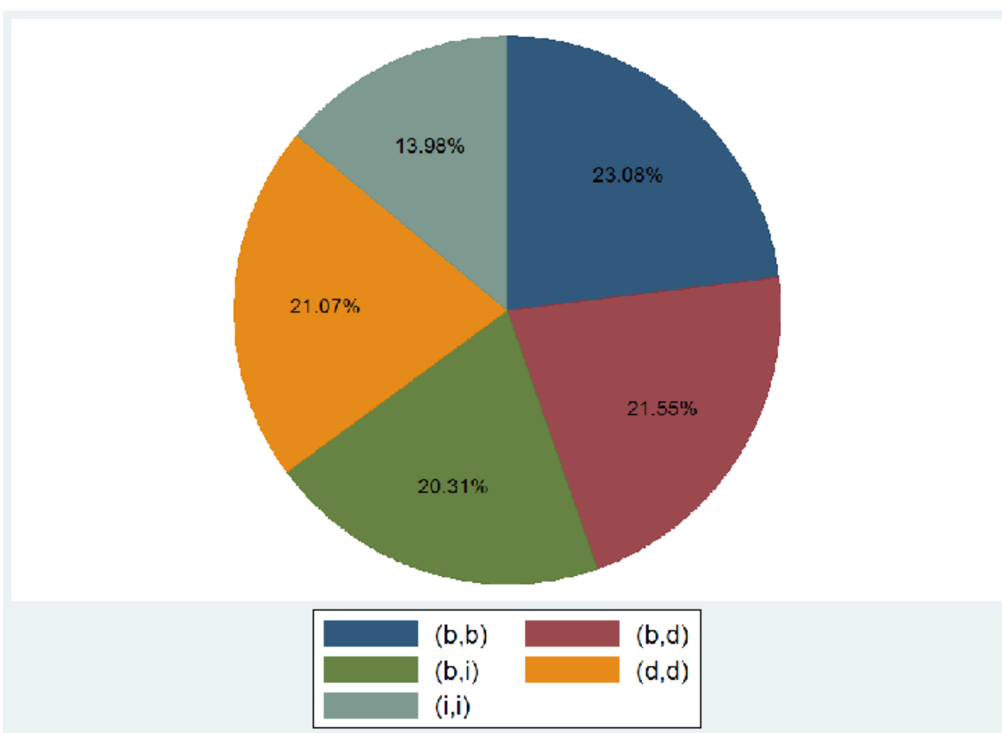


TABLE I: DESCRIPTIVE STATISTICS OF S, S^*, AND r^*; 1,044 APPS

$(S,\ S^*)$	mean r^*	st. dev. r^*	min r^*	max r^*	N
$(b,b) : r_i^*$	.028	.047	.001	.30	241
$(b,b) : r_d^*$	.030	.051	.001	.30	241
(b,i)	.010	.012	.001	.122	212
(b,d)	.010	.014	.001	.086	225
(i,i)	.011	.027	.001	.30	146
(d,d)	.012	.024	.001	.30	220
overall: r_i^*	.019	.037	.001	.30	599
overall: r_d^*	.017	.033	.001	.30	686

TABLE II: DESCRIPTIVE STATISTICS OF X; 1,044 APPS

X	Percent of Apps
game	27%
publicly traded	37%
mobile	42%
online	29%
offline	29%

the app had enough quantity demanded to be observed in comScore on both or only one of the platforms. For example, the notation (b, d) means that the app was written for both platforms but crosses the comScore threshold only on the Android platform. Descriptive statistics for the dependent variables are given in Table I.

The characteristics associated with app a (or its developer), X_a, comprise the regressors in our estimation. All our regressors are indicator variables; descriptive statistics are provided in Table II. We use information about whether the developer is publicly traded and the developers' other lines of business to indicate whether the developer is an established firm or an entrepreneur. Developers can be a mobile-only business (e.g., Rovio Entertainment, maker of Angry Birds) or have other lines of business online (e.g., Facebook) or offline (e.g., Delta, Nike, other brick and mortar stores selling physical goods or services). Publicly traded firms that have another line of business either offline or online are considered established firms. Firms that are not publicly traded and are mobile only are considered entrepreneurs. Although 54% of apps exhibit advertising and 29% of apps use in-app purchases (IAP), it turns out that publicly traded apps are almost perfectly correlated with apps with no advertising and no IAP, so we only use publicly traded in our final estimation.[7]

Our model will produce estimates of the joint distribution of the demand for apps across platforms. Some raw correlations illuminate the information in our data that will identify key parameters. In Figure

7. We did use both ads and IAP with no significant contribution in our results.

V, note that, based on S , over half of the apps multihome, but that only one-third of the apps appear to multihome in comScore based on S^*. This means that an app was able to achieve threshold reach on one platform, but not the other. Is this driven by different customer preferences on the platforms, or different marketing costs? The variation in this gap across the different observables contained in X will allow us to identify parameter values which will suggest that the latter is the stronger driver.

The correlation between r_d^* and r_i^* when both are observed ($S^* = (1, 1)$) is 0.60. When we restrict attention to those apps for which the demand threshold is raised to 1%, the correlation is 0.86. This correlation will identify ω, a structural parameter on the correlation of the joint distribution between $\widetilde{r}_i$ and $\widetilde{r}_d$.

V. Econometrics of Developer Supply

Our econometric model of developer supply needs to do four things. First, it needs to predict the probability that a particular developer publishes for either or both platforms and the success, measured as (censored) reach, the app receives there. Second, it needs to deal with the problem of selection of potential entrants. Third, it needs to accommodate both the parts of the sampling frame we know about, e.g. comScore's "panel," and the parts we do not know about, such as heterogeneity across developers. Finally, it needs to provide estimates, to the extent there is information in the data to identify them, of the main economic elements of the model. These goals lead us to a number of linked modeling decisions:

- Our sample selection rule (since we collect our own data, we have some control over this.)
- Our choice of functional form.
- Our approach to the profitability signal received by the developer.

V.A. Joint Distribution of Developer Signals and Realized App Success

Since we know the comScore sampling frame, it is easy to work out the sampling distribution of S^* and of r^* conditional on S and r. As a threshold point, the comScore sampling frame is undertaken in two separate samples, so we can work out the independent distribution on each platform. It is easy to work out the distribution of S_p^* and r_p^* conditional on r_p. comScore has a sample of 5,000 users. Let g be the number of platform p users that have the app in the comScore sample. The density of g is given by the binomial density, $f(g|r_p) = \binom{5000}{g}(r_p)^g(1-r_p)^{5000-g}$ so the probability of the event $g > 5$, can be easily calculated. Call this probability (which is one minus the binomial CDF evaluated at $g = 6$) $F_g(r_p)$. $F_g(r_p)$ is $\Pr(S_p^* = 1|r_p)$. If the distribution of r_p across apps is beta, then the sampling distribution of S_p^*

is beta-binomial. We choose our model of developer signals and realized app success to take advantage of this.

We assume that developers get signals of the potential app success $\widetilde{r} = (\widetilde{r}_i,\ \widetilde{r}_d)$ that have marginal beta distributions but that need not be independent. Our model of this is a mixture model built up from underlying beta distributions. There are three independent beta random variables, which we call q_i, q_d and q_b. The developer receives a signal

$$(\widetilde{r}_i,\ \widetilde{r}_d) = \begin{array}{ll} (q_b, q_b) & \text{with probability } \omega \\ (q_i, q_d) & \text{with probability } 1-\omega \end{array}$$

Further, we assume that $q_i \sim \text{beta}(\alpha_i, \beta_i)$ and $q_d \sim \text{beta}(\alpha_d, \beta_d)$, and that q_b is distributed beta with parameters that give it the mean and variance which are the average of those of q_i and q_d.[8] So the parameters to be estimated here are ω and α_i, β_i, which we interpret as the iOS parameters, and α_d, β_d, which we similarly interpret as the Android parameters.

Our second modeling choice is the relationship between the developer's pre-entry signal and post-entry success. We reject the simplest model in which these are perfectly correlated based on industry facts described above, such as the frequency of apps introduced with a large marketing push in the app stores, that are not ultimately successes. Our specific model of less-than-perfect dependence between signal and ultimate success draws the distinction between the reach the app would have if it were visible to all customers by being on the top list in the app store, $\widetilde{r}_p$, and the reach it ultimately achieves, r_p.

Not all apps can appear on the app store's top list, and we model the probability that an app appears on the top list as $A(\widetilde{r}_p, \tau)$. Here $\widetilde{r}_p$ is the fraction of platform users that would choose the app if they knew of it and τ is a parameter. So $A()$ tells us the efficiency of the app store at matching, and τ is a parameter that indexes the efficiency of the app store at matching. Similarly, we introduce another random variable, $\overline{r}_p$, the reach the app would gain if it were relegated to the depths of the app store rather than being on the top list. We assume that this is a "shrunk" version of $\widetilde{r}_p$, i.e. that $\overline{r}_p$˜beta and that the parameters of this beta make $E[\ \overline{r}_p] = \delta E[\ \widetilde{r}_p]$ and $var[\ \overline{r}_p] = \delta var[\ \widetilde{r}_p]$, with δ a new parameter to be estimated.

These assumptions mean that the effective demand for the app, r_p, is given by

$$r_p = \begin{cases} \widetilde{r}_p \text{ with probability } A(\widetilde{r}_p, \tau) \\ \overline{r}_p \text{ otherwise }. \end{cases}$$

The variable r_p determines the developer's potential profits on platform p, and if the developer does enter, r_p is the random variable censored to determine S_p^* and, r_p^*.

Therefore, the app is written for platform p if and only if the following condition is satisfied:

8. We assume that $\mu_b = \frac{\alpha_b}{\alpha_b + \beta_b} = \frac{\mu_i}{2} + \frac{\mu_d}{2}$ and $\sigma_b^2 = \frac{\sigma_i^2}{2} + \frac{\sigma_d^2}{2}$ in our estimation.

$$\mathbb{E}[r_p|\widetilde{r}_i, \widetilde{r}_d] = \mathbb{E}[\widetilde{r}_p A(\widetilde{r}_p, \tau) + (1 - A(\widetilde{r}_p, \tau))\overline{r}_p|\widetilde{r}_i, \widetilde{r}_d] \geq \kappa_p. \tag{4}$$

V.B. Parameters and Likelihood

The parameters of the model, which together we call θ, are $\alpha_i, \beta_i, \alpha_d, \beta_d$, ω, τ, δ and κ_p. All of these parameters will, in some specifications, vary with the regressors X_a. Above, we showed the conditions for observing r_p^*, and the sampling distribution of r_p^* conditional on r_p. Similarly, we showed the probability that S_p^*,$= 1$ conditional on r_p. Finally, we showed the event determining S, which is a crossing condition for *beta* random variables. Under our assumptions, the distribution of r_p is beta, so we can calculate (using the beta binomial distribution) the likelihood of r^* and S^* and the probability of S. The joint distribution of r^*, S^* and S is the likelihood $L(S,\, S^*,\, r^*|X_a,\, \theta)$. We can calculate it in closed form given our assumptions, of which the most important for this purpose is the beta functional form.

It is not hard to see how our model is identified and the role of the functional form of the distribution in it. The distribution of $(r_i,\ r_d)$ has *beta* marginals that depend on $\alpha_i, \beta_i, \alpha_d, \beta_d$, ω, τ, and δ. The distribution of $(r_i,\ r_d)$ are easily recovered by predicting the sample distribution of $(r_i^*,\ r_d^*)$ conditional on observing them. To get from the distribution of $(r_i,\ r_d)$ back to the underlying distribution of $(\widetilde{r}_i,\ \widetilde{r}_d)$ and the mixing parameters we use the probability of the events $(S,\ S^*)$ shown in Figure V.

The structure of the dependence among $(\widetilde{r}_i,\ \widetilde{r}_d)$ appears limiting, because it is a mixture model, but for predicting S and r^*, this is not an issue. The event $S_{p'} = 1$ occurs not only when $(\widetilde{r}_i,\ \widetilde{r}_d) = (q_b,\ q_b)$, but also when both of the latent variables $\widetilde{r}_i$ or $\widetilde{r}_d$ takes a large value.

V.C. Sample

Since we collect our own data from primary sources, we have some control over the definition of our sample. It is not possible to design a sample which is entirely free of selection bias: app developers who have had an idea but who have failed to publish their app, for example, cannot in principle be observed. We can, however, minimize the degree to which sample selection bias correction leads us to introduce new assumptions.

Our sample consists of apps with $S^* \neq (0, 0)$, that is apps which appear in comScore on at least one platform. The rationale for this is simple, if novel. Given that we only observe the censored variable r_p^* if $S_p^* = 1$, we must introduce a model of the event $S_p^* = 1$ in order to deal with the censoring. Under our sampling scheme, observed potential entrants into platform p' (the "other" platform) are those who appear

TABLE III: App Supply States

d, i	$S_i = 0$	$S_i = 1$	$S_i^* = 1$
$S_d = 0$		r_i	r_i^*
$S_d = 1$	r_d	r_d, r_i	r_i^*
S_d^*	r_d^*	r_d^*, r_i	r_d^*, r_i^*

in the comScore data on platform p ("this" platform.) With an explicit model of the app appearing on "this" platform, we can correct for the selection bias in our treatment of entry into the "other" platform. Of course, we treat both platforms symmetrically and examine the joint distribution of S, S^*, and r^*.

The two sets of two dummy variables S and S^* define 9 possible states (combinations of whether an app is available on each platform and whether it has achieved the demand threshold to be observed in comScore for each platform). Because of our sampling scheme, however, we only observe 5 of these states. These are the 5 cells in Table III which contain at least one r_p^*. The reach for each state is presented in each cell of the table. Since this sample definition is conditional on the dependent variable of app supply to platforms, we must also redefine the likelihood to adjust for this conditioning, a topic we address in a moment.

Entry models generally need to condition on something in order to define the set of potential entrants. The problem is that firms that have not entered any market do not exist and that data on them cannot be gathered. A variety of solutions have grown up to deal with this. Berry (1992), and many papers following in that tradition, define the set of potential entrants into a particular market by observing firms that have already entered related markets. In his example, potential entrants into a particular airline city pair market are firms serving other city pair markets already. This sample definition conditions on the lagged dependent variable in an adjacent market. One could hope that the double attenuation that comes both from lagging and from looking at an adjacent market renders any sample selection bias small (the most common solution) or one could attempt to correct for sample selection bias. Another solution is to avoid defining a set of potential entrants, but instead examine a fixed set of market niches each of which could (or could not) be filled by a particular entrant. Bresnahan and Reiss (1991) use this approach, as does Mazzeo (2002) who looks at entry in the categories of high-quality motels and low-quality motels. The Bresnahan-Reiss approach only works when one can define the market niches a priori, so we cannot use it. The Berry approach, with a correction for the sampling process, works for us because we observe a parallel market, i.e., for entry into iOS we look at hit apps on Android.

Our model can be understood as having many of the features of a "Type II" Tobit. This familiar model is written $y_2 = \left\{ \begin{matrix} y_2^* \text{ if } y_1^* > 0 \\ 0 \text{ if } y_1^* \leq 0 \end{matrix} \right\}$, where y_2 is a variable, like reach on one platform for app a for us, that is only observed if an event, labeled in the Tobit as $y_1^* > 0$, occurs.

A simplified version of our model would have exactly this structure. Suppose we were only studying entry into platform d, and we gather our sample of firms that have entered and been successful on platform i. Then we would let $y_2^* \equiv r_D$ and introduce $y_1^* \leq 0$ as a model of sample selection. If these two random variables (demand on platform d and entry and success on platform i) are correlated, as one would expect, then we would use the Type II Tobit to deal with the relevant sample selection problem.

This simplified version is how we solve a classic econometric problem in models of entry. If the set of potential entrants into one market are actual entrants in another market, and if profits in the two markets are correlated, there is a sample selection problem.

We use this same sample selection logic but embed it in the structure of our model, which involves a few changes in the specifics. First, our model is symmetric across the two platforms, so that there are two y_2^* variables, and we have different conditions for observing the two y_2 variables. We have two variables like y_2: r_d and r_i are both observed only conditional on events. Second, in our model, $y_1^* > 0$ (the condition for observing r_d or r_i) is a compound event, depending not only on the realization of reach but also on the app being included in our sample. We observe r_d if $r_d^* > .001$. Third, we observe not only the truncated reach, but also the fact of writing for the platform or not; these events are observable whenever the app is part of our sample. None of these complications changes the nature of our approach to sample selection profoundly.

We choose a symmetric (across platforms) sample selection rule, and thus study entry into (and success on) the markets defined by both platforms simultaneously. We have both economic and econometric reasons for taking this approach. From an economic perspective, our platform focus leads us to studying multihoming, i.e., entry into both platforms, which requires the symmetric treatment. From an econometric perspective, modeling both platforms at once is our best identification strategy. We have no instruments for the "selection equation," so we need to put economic restrictions on the conditioning event. In our approach, each conditioning event has a full economic model, since it is part of our dependent variable.

V.D. Sampling Correction

We start from the universe of potential app developers who have an idea for an app. We have already defined the unconditional likelihood, $L(S,\ S^*,\ r^*|X_a,\ \theta)$. We have already shown how to calculate the probability of the events $S_p^* = 1$. Using the same logic, we calculate the probability, $F_S(S,\ S^*,\ r^*|X_a,\ \theta)$ that either $S_d^* = 1$ or $S_i^* = 1$ or both. Then we maximize the conditional likelihood

TABLE IV: Model Estimates of Reach Moments

	Mean Reach on Android		Mean Reach on iOS	
	Potential	Realized	Potential	Realized
In Sample	0.0310	0.0076	0.0293	0.0079
In Population	0.0294	0.0070	0.0280	0.0076
	(0.0064,0.0383)	(0.0036,0.0089)	(0.0067,0.0373)	(0.0033,0.0089)

$$L_C(S, S^*, r^*|X_a, \theta) = L(S, S^*, r^*|X_a, \theta)/F_S(S, S^*, r^*|X_a, \theta) \tag{5}$$

VI. Results

In Table IV, we present the mean estimated potential reach $\widetilde{r}_p$ and mean estimated realized reach r_p for apps on Android and iOS based on our model. The first line of results shows these calculations using sample frequencies as weights. The second line of the results shows these calculations when we weight using the implied population frequencies calculated using Bayes law. Bootstrapped confidence intervals are in parentheses below the calculations.

The difference between the sample and population results reflects the effect of accounting for entry selection. As expected, when we compare the first and second rows within each of the four columns, the sample reaches are higher than the population reaches, since the sample does not consider the distribution of regressors in the apps which did not reach the comScore observation threshold.[9] Those would have been apps with lower reach, and thus they bring down the mean reach in the population estimates. Henceforth, we focus on the population estimates, since we are interested in the implications of our estimates for all developer activity.

The first result we have comes from comparing the respective reaches of Apple to Android. There is strikingly little difference between the expected potential reach on Android compared to iOS (0.0294 vs. 0.0280) and the expected realized reach on Android compared to iOS (0.0070 vs. 0.0076).[10] This implies that there is very little difference from the developer's perspective in terms of the type of customers they face on iOS and Android: regardless of which platform they enter, they are likely to get the same demand response from the users on that platform. We will examine later in this section whether this varies with app and developer heterogeneity.

9. Tests of signficance of differences forthcoming.
10. Tests of signficance of differences forthcoming.

TABLE V: Model Estimates in Population

	Mean Realized Reach on Android	Mean Realized Reach on iOS
None	0.0070	0.0076
	(0.0036,0.0089)	(0.0033,0.0089)
Written for Android	0.0091	0.0085
	(0.0052,0.0118)	(0.0046,0.0102)
Written for iOS	0.0089	0.0100
	(0.0060,0.0119)	(0.0059,0.0122)

When we compare the difference between potential and realized reach within each of the platforms (0.0294 vs. 0.0070 on Android and 0.0280 vs. 0.0076 on iOS), we observe a substantial difference between these numbers.[11] This suggests that there is value to allowing our model to estimate two separate reaches, and if we interpret this gap as representative of marketing costs, it confirms our findings from industry interviews and in the rankings data that marketing is a significant cost to successfully reaching demand.

We now focus our attention on the population mean estimated realized reach estimates, since this is what determines app profitability. We examine mean estimated r_p conditional on having entered the iOS or Android markets. The first row of Table V replicates the second line of Table IV, representing the estimated mean reach taken over the entire population of types. The second row of Table V presents the estimated mean reach if we focus on those apps which were written for Android. The first column of estimates shows the reach for those apps on Android, and the second column shows the reach for those apps on iOS. The third row of Table V presents the estimated mean reaches for apps which were written for iOS. Again, the first column shows the estimated reach for those apps on Android, and the second column shows the estimated reach for those apps on iOS. Bootstrapped confidence intervals are in parentheses below all figures.

If we examine the reach on Android in the first column, we see that conditioning on the fact that this app was written for Android makes the expected realized reach much higher than in the overall population, as expected (0.0091>0.0070).[12] The equivalent comparison for Apple can be found by looking at the row "Written for iOS" and the last column of the table, "Mean Realized Reach on iOS". Again, we see that knowing that an app was written for iOS implies a much higher estimated mean reach (0.0100>0.0076). These results themselves simply reflect developer rationality, but we can learn much by comparing them quantitatively to other statistics.

One such comparison looks at how the condition that an app was written for *Android* affects the estimated reach of that app on *iOS* (second row and last column of Table V). In other words, what is the expected realized reach that would arise from multihoming on iOS for an app knowing only that it was written for

11. Tests of signficance of differences forthcoming.
12. Tests of signficance of differences forthcoming.

Android? We see that compared to the overall population, the reach is higher for apps which have already been written on Android (0.0085>0.0076).[13] Analogously, if we look at the last row and first column of Table V, we see that an app written for iOS is going to have much higher estimated reach on Android than in the overall population (0.0089>0.0070). What is striking about these numbers is the comparison to the figures we examined in the last paragraph. Predicted demand for an app on Android conditional on being written for iOS is quite similar to predicted demand for an Android app conditional on being written for Android itself. The symmetric result holds for an app on iOS. Our estimates are revealing large positive dependence between the demand for an app on iOS and the demand for the same app on Android.

Table V allows us to calculate an index number that suggests that most of the contribution from app availability to the attractiveness of either platform to users comes from multihoming apps, not from apps that uniquely choose that platform. Recall from our economic model of users that the index of the platform value to users is the sum of reaches of all apps available on that platform, $\sum_{a \in N_p} r_{pa}$. The value of multihoming apps can therefore be represented by the ratio of the sum of r_p for all apps written for p', denoted $\sum_{a \in N_{p'}} r_{pa}$, to $\sum_{a \in N_p} r_{pa}$, which is equivalent to the ratio of the mean r_p written for p' to mean r_p if $N_{p'} = N_p$, which is approximately true in our setting. So for Android, the percent contribution of iOS apps if they were launched on Android to Android's value to users is $0.0089/0.0091 = 98\%$. For iOS, the percent contribution of Android apps if they were launched on iOS to iOS's value to users is $0.0085/0.0100 = 85\%$. The contribution of multihomers to platform value is slightly higher on Android than on iOS, but in both cases the majority of both platforms' values comes from multihomers.

In Table VI, we present the distribution of mean realized reach on each platform broken out by different observable types of developers and apps captured by X. We also calculate the difference between those reaches to test whether there are differences in preferences for platforms connected to these observable types. The last column indicates the number of apps in each category from our sample. Confidence intervals for the preference differences are reported underneath in parentheses. In all cases, the differences are not statistically different. Indeed, if we examine the parameter estimates for each of these cases reported in Tables **??**-X, we see that the estimated threshold reach κ_p over which a developer will write to a platform is very similar in all cases across platforms, so even the decision rule for both platforms is almost identical.[14] This contradicts the explanation that strong platform preferences divide the developers. The developers have no particular reason to favor one of these platforms over the other except to the extent that one platform has more users than the other: in that case, the equal reaches on both platforms would lead to a higher absolute number of customers on the larger platform.

13. Tests of signficance of differences forthcoming.
14. Tests of signficance of differences forthcoming.

Table VI also shows that the developers in the observable categories with the highest mean reach (Non-Games, Online and Offline) tend to also have the smallest preference differences between the two platforms. Furthermore, estimates of ω, the correlation on the joint distribution between potential reaches $\widetilde{r}_i$ and $\widetilde{r}_d$, are shown in Tables VIII and X to be particularly high for Publicly Traded (whether Online or Offline) developers. Note that Non-Games, Online and Offline, and Publicly Traded are all characteristics of established firms. The combination of highly correlated and high reach on both platforms and small preference differences between platforms creates a strong incentive for these apps to multihome. As a result, consumers can access these applications, which they find most attractive, on either platform. In contrast, the Mobile Only firms (most of whom are Not Publicly Traded), reflecting entrepreneurs, have the lowest reach on all platforms and the lowest estimated ω.

Our results are consistent with the idea that more established firms avoid the marketing costs of reaching customers because they already have a relationship with their current installed base of customers. They are therefore not encumbered by the competition to rank highly on the collaborative filters. The costs of the app store fall heavily on firms that are trying to acquire new customers in the mobile world (e.g. mobile entrepreneurs) and much less for those for whom the mobile world is an extension of existing customer relationships (e.g. banks).

In results not reported in tables, we check the stability of our results over time by running our model on data from September, 2012. All results are similar. We interpret this as confirming that there has been little change in developer incentives over the relevant time scale. Of course, checking over a much longer period of time might reveal movements in developer preferences.

VII. Conclusion

We have estimated a model of developer platform choice on a new dataset of mobile app developers' smartphone supply decisions. We examine the decision to supply for either or for both of the dominant platforms. Our model treats supplying for a platform much like models of entry treat serving a market: a platform will be supplied with an app only if expected demand for the app on that platform is high enough to generate variable profits that cover the incremental fixed costs of writing and marketing. Fortunately for identification, we have some outcome variables in addition to developers' app supply decisions. For apps that are at least moderately successful, we also observe quantity demanded in a sample of users.

Our model has a novel element that we hope is useful in entry studies generally. We show how to correct for the unavoidable problem that the set of potential entrants into markets is either unobserved or selected. In our application, the quantitative importance of selection is high, as we see large differences in estimated

TABLE VI: Expected Differences in Platform Preferences (Android - iOS) and Mean Estimated Realized Reach

Lines of Business	Pref Difference	Android	iOS	N
Games, Not Publicly Traded				
Online	-.0021	.0052	.0073	48
	(-.0080,.0012)			
Mobile Only	-.0003	.0037	.0040	177
	(-.0024,.0011)			
Offline	-.0025	.0058	.0083	22
	(-.0065,.0001)			
Games, Publicly Traded				
Online	-.0019	.0059	.0078	24
	(-.0090,.0021)			
Mobile Only	.0015	.0036	.0021	3
	(-.0035,.0049)			
Offline	-.0023	.0064	.0087	53
	(-.0083,.0009)			
Non-Games, Not Publicly Traded				
Online	-.0007	.0092	.0099	151
	(-.0050,.0044)			
Mobile Only	.0014	.0062	.0049	257
	(-.0011,.0055)			
Offline	-.0011	.0083	.0094	76
	(-.0048,.0033)			
Non-Games, Publicly Traded				
Online	-.0004	.0163	.0167	80
	(-.0081,.0054)			
Mobile Only	.0053	.0086	.0033	1
	(-.0025,.0114)			
Offline	-.0003	.0102	.0105	152
	(-.0069,.0037)			

TABLE VII: Model Parameter Estimates (GAMES, No PT)

Parameter	Android	iOS
τ	.87	
	(0.39,1.00)	
δ	.19	.21
	(0.22,0.96)	(0.12,0.90)
Online Firm		
α	1.46	0.94
	(0.22,2.33)	(0.20,1.86)
β	62.2	31.8
	(28.0,81.5)	(17.9,61.5)
κ	.0054	.0072
	(.0026,.0085)	(.0036,.0128)
ω	.56	
	(.10,.96)	
Mobile Only Firm		
α	1.08	0.69
	(0.08,1.54)	(0.07,1.79)
β	62.4	41.8
	(31.1,83.2)	(29.1,73.9)
κ	.0033	.0035
	(.0006,.0040)	(.0008,.0052)
ω	.11	
	(.20,.68)	
Offline Firm		
α	2.28	2.62
	(0.04,2.47)	(0.14,3.70)
β	86.4	75.6
	(30.8,102.1)	(28.4,101.0)
κ	.0052	.0074
	(.0006,.0071)	(.0014,.0110)
ω	.61	
	(.26,.91)	

TABLE VIII: Model Parameter Estimates (GAMES, PT)

τ	.87	
	(0.39,1.00)	
δ	.19	.21
	(0.11,0.96)	(0.12,0.90)
Online Firm		
α	1.14	0.46
	(0.12,4.48)	(0.11,2.26)
β	45.4	14.9
	(24.7,66.2)	(6.50,50.2)
κ	.0091	.0102
	(.0046,.0210)	(.0062,.0198)
ω	.88	
	(.60,1.0)	
Mobile Only Firm		
α	0.76	0.20
	(0.04,2.83)	(0.03,1.54)
β	45.6	24.9
	(29.5,76.4)	(14.2,62.2)
κ	.0069	.0065
	(.00153,.0184)	(.0022,.0171)
ω	.43	
	(.35,.96)	
Offline Firm		
α	1.96	2.13
	(0.30,3.17)	(0.33,3.15)
β	69.6	58.7
	(33.2,83.9)	(16.9,80.3)
κ	.0089	.0103
	(.0038,.0223)	(.0056,.0209)
ω	.93	
	(.69,.97)	

TABLE IX: Model Parameter Estimates (non-GAMES, No PT)

τ	.87	
	(0.39,1.00)	
δ	.19	.21
	(0.11,0.96)	(0.12,0.90)
Parameter	Android	iOS
Online Firm		
α	1.27	0.96
	(0.22,2.81)	(0.23,2.55)
β	33.3	25.8
	(18.7,49.6)	(13.4,45.4)
κ	.0072	.0079
	(.0050,.0122)	(.0045,.0133)
ω	.49	
	(.00,.87)	
Mobile Only Firm		
α	0.89	0.70
	(0.16,1.96)	(0.15,1.79)
β	33.4	35.8
	(12.6,65.4)	(26.3,63.5)
κ	.0050	.0042
	(.0030,.0092)	(.0025,.0064)
ω	.05	
	(.00,.64)	
Offline Firm		
α	2.09	2.64
	(0.28,2.36)	(0.34,3.97)
β	57.4	69.6
	(23.9,70.9)	(26.9,80.0)
κ	.0070	.0081
	(.0037,.0111)	(.0044,.0112)
ω	.54	
	(.17,.89)	

TABLE X: Model Parameter Estimates (non-GAMES, PT)

τ	.87	
	(0.39,1.00)	
δ	.19	.21
	(0.11,0.96)	(0.12,0.90)
Parameter	Android	iOS
Online Firm		
α	0.95	0.47
	(0.10,4.97)	(0.08,3.20)
β	16.5	8.95
	(6.21,45.3)	(4.42,41.1)
κ	.0109	.0109
	(.0081,.0226)	(.0075,.0215)
ω	.81	
	(.49,.96)	
Mobile Only Firm		
α	0.57	0.22
	(0.05,3.96)	(0.03,2.42)
β	16.6	19.0
	(3.55,55.2)	(6.69,52.2)
κ	.0087	.0072
	(.0043,.0200)	(.0030,.0188)
ω	.37	
	(.10,.89)	
Offline Firm		
α	1.77	2.15
	(0.30,3.72)	(0.37,3.30)
β	40.6	52.7
	(22.3,50.1)	(19.5,57.2)
κ	.0107	.0111
	(.0074,.0239)	(.0076,.0226)
ω	.86	
	(.29,.95)	

developer profitability between the selected sample of potential entrants and the much larger population.

We find that there are dramatic differences in app profitability by observable developer type and app type. From an economic growth perspective, the most important of these is the difference between entrepreneurial app developers and established firms. Since the mobile app platforms provide the vast bulk of the technical inputs needed to create a working system, many forecast, correctly, that there would be a wave of entrepreneurship in mobile apps. Many also forecast that this wave of entrepreneurs would disrupt existing businesses on a massive scale. Perhaps so someday, but at this stage the supply of mass market mobile apps is overwhelmingly from established firms. Based on our many industry interviews, we attribute much of the difference between firm types to marketing costs. (Our estimates can only pin it on fixed costs, not on the form of fixed costs.) It appears that the high costs of finding customers for new firms – some of which can be attributed to the considerable difficulty of matching buyer to seller on the platform-supplied app stores – are limiting the role of entrepreneurship.

The platform race on smartphones in the US has been approximately tied for a surprisingly long time. Our results provide an explanation of this otherwise quite surprising fact. As a threshold empirical result, we see that (1) there is little difference in expected profitability for developers between the two platforms. This means that an equilibrium with divided platform choices exists. Such equilibria are typically not stable, and platform markets tend to tip from them. However, our other results explain why this market does not tip: (1) The distribution of app attractiveness to consumers is skewed, with a small minority of apps drawing the vast majority of consumer demand. (2) Apps which are highly demanded on one platform tend also to be highly demanded on the other platform. (3) These highly demanded apps have a strong tendency to multihome, writing for both platforms. As a result, the presence or absence of apps offers little reason for consumers to choose a platform. A consumer can choose either platform, and have access to the most attractive apps. This undercuts the mechanism by which the platform market might tip. What would ordinarily be the unstable platform market equilibrium, with an approximate tie between iOS and Android, is rendered stable. This result, which we came to empirically, does not appear in the theoretical literature.

VIII. Bibliography

Augereau, Angelique, Shane Greenstein, and Marc Rysman. 2006. "Coordination vs. Differentiation in a Standards War: 56K Modems." The RAND Journal of Economics, 37(4): 771-88.

Berry, S.T. (1992), "Estimation of a Model of Entry in the Airline Industry", Econometrica 60(4): 889-917.

Berry, S.T., and E. Tamer (2006), "Identification in Models of Oligopoly Entry", xxxx

Berry, S.T., and J. Waldfogel (1999), “Social Inefficiency in Radio Broadcasting”, Rand Journal of Economics, 30(3): 397-420.

Boudreau,KJ and A Hagiu Platform rules: Multi-sided platforms as regulators

Bresnahan, T.F. and P.C. Reiss (1990), “Entry in Monopoly Markets”, Review of Economic Studies 57: 57-81.

Bresnahan Davis Yin XXX

Brown, J. and J. Morgan (2009) "How much is a Dollar Worth? Tipping versus Equilibrium Coexistence on Competing Online Auction Sites,” Journal of Political Economy, 117: 668-700.

Cantillon, E. and P. Yin, "Competition Between Exchanges: Lessons from the Battle of the Bund," 2013

Church, Jeffrey, and Neil Gandal. 1992. "Network Effects, Software Provision, and Standardization." Journal of Industrial Economics, 40(1): 85-103.

Ciliberto, F., and E. Tamer (2003), “Market Structure and Multiple Equilibria in the Airline Industry”, xxx

Corts, Ken, and Mara Lederman. "Software Exclusivity and Indirect Network Effects in the US Home Video Game Industry." International Journal of Industrial Organization. 259-70.

Damiano, Ettore, and Hao Li. 2008. "Competing Matchmaking." Journal of the European Economic Association, 6(4):789-818.

Davis, J., Y. Muzyrya, and P. Yin, "Killer Apps," 2013.

Davis, P. (1997), “Spatial Competition in Retail Markets: Motion Theaters”, Rand Journal of Economics.

Dubé, Jean-Pierre, K. Sudhir, Andrew Ching, Gregory S. Crawford, Michaela Draganska, Jeremy T. Fox, Wesley Hartmann, Günter J. Hitsch, V. Brian Viard, Miguel Villas-Boas "Recent Advances in Structural Econometric Modeling: Dynamics, Product Positioning and Entry" Marketing letters ’05

Ellison, Glenn and Drew Fudenberg. 2003. "Knife-Edge or Plateau: When Do Market Models Tip?," Quarterly Journal of Economics, 118(4), 1249-1278.

Greenstein, Shane, and Marc Rysman. 2007. "Coordination Costs and Standard Setting: Lessons from 56K Modems." In Standards and Public Policy, ed. S. Greenstein and V. Stango. Journal of the European Economic Association, 123-59. Cambridge University Press.

Lee, Robin. 2013. "Vertical Integration and Exclusivity in Platform and Two-Sided Markets."

Mazzeo, M. (2002), “Product Choice and Oligopoly Market Structure”, Rand Journal of Economics 33(2): 1-22.

Rysman, M. (2004), “Competition Between Networks: A Study of the Market for Yellow Pages”, Review of Economic Studies 71:

483-512.

Rysman, Marc, The Economics of Two-Sided Markets 111(2): 353-403.

Rysman, Marc. 2007. "An Empirical Analysis of Payment Card Usage." Journal of Industrial Economics, 55(1): 1-36.

Seim, K. (2004), “An Empirical Model of Firm Entry With Endogenous Product-Type Choices”, Rand

Sinkinson, Michael Pricing and Entry Incentives with Exclusive Contracts: Evidence from Smartphones xxx

Small, K. A. and H. S. Rosen. 1981. "Applied Welfare Economics with Discrete Choice Models," Econometrica, 49(1):105-130.

Zhu, Ting, Hongju Liu and Pradeep K. Chintagunta. 2011. “Wireless Carriers’ Exclusive Handset Arrangements: An Empirical Look at the iPhone,” NET Institute

working paper.