Trade-offs of Using Middleware: An Innovation Enabler and Creativity Constraint

Milan Miric*

mmiric@marshall.usc.edu - USC Marshall

Hakan Ozalp

h.ozalp@vu.nl - Vrije Universiteit Amsterdam

Erdem Dogukan Yilmaz

ErdemDogukan.Yilmaz@unil.ch - HEC Lausanne

ABSTRACT

A critical determinant of the success of digital platforms is the availability of third-party complements. In order to foster the creation of such complements, platforms often rely on boundary resources: software tools and libraries that support and lower the costs of developing complementary innovations. However, while such boundary components are increasingly widespread, little is understood about how these components shape subsequent innovation outcomes and the overall platform generativity. Here, we focus on a particularly important form of boundary resource for digital platforms: Middleware components for software development. We theorize how middleware may shape both the novelty of products being created and their subsequent market value, and propose a number of mechanisms for these effects, for which we provide evidence. We test for these different mechanisms by looking at the development of console games during 6th and 7th generation of gaming consoles. We find that the introduction of licensed middleware such as third-party game and graphics engines lead to the creation of less novel, but more commercially successful products. We attribute this to the fact that middleware allows firms to reallocate resources from developing basic functionality (such as programming how the graphics will be rendered for the target platform), to other elements that make games more valuable and successful on the market (such as story or game mechanic innovation). Our results have implications for how we think about the impact of boundary resources such as middleware on digital platforms, and recombinant innovation more broadly.

Keywords: Two-sided platforms; modularity; innovation tools; recombinant innovation; boundary resources

* We acknowledge comments and suggestions from participants of seminars and presentations at AoM 2018 PDW on Direction of Innovation, CMU-Temple 2018 Conference on Digital Marketing and Machine Learning, Vrije Universiteit Amsterdam, 2019 AoM MOC Division Frontiers Conference, Munich Summer Institute 2019, and DRUID19 Copenhagen.

Note: Due to page limit of the submission, Appendices are not included. They are available from the authors upon request.

1. INTRODUCTION

Platforms and platform-based ecosystems are increasingly a common way of organizing, in part because platforms have the ability to offer customers both scale and scope of offerings created by third-parties in the surrounding platform ecosystem (Gawer, 2014; Jacobides et al., 2018). These platform-based ecosystems are highly generative and allow the creation of new innovations "through unfiltered contributions from broad and varied audiences" (Zittrain 2008, p. 70; Yoo et al., 2010). This generativity on the part of third-party complementors is often fostered by tools and resources the platform provides such as Software Development Kits (SDKs) and Application Programming Interfaces (APIs), as well as the policies and norms that platform determines to govern the activities of participating complementors (Ghazawneh & Henfridsson, 2013; Eaton et al., 2015).

While the platform itself can provide many of these tools (e.g., SDKs for software development), there is often space for a layer of independent companies that specialize in the development of enabling technologies and resources. For example, in the case of Wordpress, a website publishing platform, external APIs can be optionally used and combined with the platform- owned, and generally compulsory, APIs to create new plug-ins (complementary products) that extend the functionality of Wordpress (Yoo, 2019). Similarly, in console gaming platform-ecosystem, third- party development tool and middleware providers are crucial in facilitating the development of complementary products (Evans, Hagiu, and Schmalansee, 2006, Corts and Lederman, 2009). In mobile app development, an additional layer of optional third-party tools, such as those simplify cross-platform app development (Kang, Aaltonen, and Henfridsson, 2019), have been developed in addition to platform provided API and toolkits. As these examples suggest, external tools and resources, although optional to adopt, can be important to the generative nature of platform-based ecosystems. Yet, little is understood about how the existence of these tools and resources shapes software development in these platforms (except the work by Yoo, 2019).

When platform owners allow the use of these enabling tools and resources, or *middleware*[1] components as referred in our setting, their goal is to often foster the generativity of their platforms. This is consistent with the core arguments behind why building on a generally applicable

[1] The term is broadly and commonly used in the video game industry, covering a range of tools from "proper middleware" that acts as the middle layer software between audio/input hardware, game code, and the output, to the complete package of game development editor and IDE with integrated middleware for various functionalities, also known as (and again, colloquially used), game engines. In this study, we are focusing only on "major" middleware that cover broader (game engine) or major (graphics rendering engine or physics engine) functions in the game which has a distinctly different impact on the organization for the development and its associated outcomes. See: http://twvideo01.ubm-us.net/o1/vault/gdc04/slides/impact_of_middleware.pdf

and modular technology can foster generativity and innovation: (i) Tools simplify the complexity of development into simple to use interfaces, for example separating the laborious task of coding from creative design tasks, that may allow the products to be more differentiated (Parnas, 1972; Baldwin and Clark, 2000); (ii) they allow fast prototyping and development through experimentation (von Hippel and Katz, 2002), and; (iii) provide module libraries with most used basic blocks in product development (Garud and Kumaraswamy, 1995; von Hippel and Katz, 2002). These three factors, via reuse and modularity, allow leveraging "economies of substitution" (Garud and Kumaraswamy, 1995; Schilling, 2002), which lowers cost of design of new innovations through partial retention of existing components rather than designing the new innovation from scratch (Garud and Kumaraswamy, 1993). However, once platform companies allow such middleware to be used by complementors on their platform[2], they have much less control regarding how developers choose to use these technologies and when they will be used. Therefore, if building on these tools: (i) makes it easier for complementors to imitate successful designs (Ethiraj and Levinthal, 2004); (ii) enables lower quality entrants to flood the market through the lower costs of complementary product development (Boudreau, 2019); (iii) limits reaching greater functionality through integrated (i.e., non-modular) design between the platform complement and the platform, known as synergistic specificity (Schilling, 2000; Baldwin, 2008; Tiwana, 2018)[3]; and (iv) puts boundaries on what can be developed as tools fixate higher order design hierarchies (Clark, 1985) and therefore imposes costs for developing novel innovations that are beyond the proximate application area of the tool (Arora, Gambardella, and Rullani, 1997; Bresnahan and Gambardella, 1998), then these tools may lead to products which are less novel and potentially less attractive to consumers.

In this paper, we explore whether the use of these tools or boundary resources, represented by middleware components such as game engines in our video game context, is associated with product novelty and commercial success. We conceptualize the use of these middleware components as a modular technology that a variety of innovations (i.e., complements) can be built "on top of" it. We present competing hypotheses for how middleware may lead to the creation of:

[2] In our setting, middleware is provided only if the platform owner explicitly provides access (Boudreau, 2010), hence, middleware provision is subject to similar governance mechanisms of the platform owner for complementors. Video game console owners provide proprietary technical information to middleware providers so their tools can abstract low-level ("hardware level") details and then middleware providers can license these tools to complement developers by lump sum/variable fees. In exchange, middleware providers themselves enter a licensing agreement in which they are screened by the platform owner (for the initial provision of the middleware and for each update), and pay lump sum and/or variable fees to the platform owner to be on the platform.

[3] Tiwana (2018) provides conceptually related "synergy" arguments across literatures. Among these definitions, our synergy definition is best captured by Schilling's (2000) synergistic specificity, as defined in the article, however also highly overlaps with other proximates, such as module specificity by (Baldwin, 2008), which is defined as: "Degree to which one system is uniquely customized to another system" (p. 170).

(i) More or less novel products; and (ii) more or less commercially successful products. We test these competing hypotheses using data on console-based video game setting that covers the set of new video game titles launched between 2000 and 2009 on the Sony (Playstation 2 and Playstation 3) and Microsoft (Xbox and Xbox 360) platforms. This is the period when middleware components were first introduced on these platforms. We track whether complementary products used middleware components or not, how of this choice relates to novelty and market performance of these complementary products. We find that the use of middleware components is associated with products that are less novel but achieve greater commercial success. We find evidence that this is both a function of the technologies themselves, but also of the companies that choose to use middleware and specialize in middleware based development.

This paper names a number of contributions. First, our central contribution to the platform-based ecosystem literature is to be among the first papers to explore the implications of such tools and external boundary resources on the generativity of the platform. Additionally, we also contribute to related platform-level issues: First, our results imply that the use of external boundary resources may actually be associated with complements that are less novel at the platform level, which goes beyond the previous study on this topic which found that external boundary resources increase generativity (Yoo, 2019). Our results suggest that the availability of enabling technologies such as middleware components can allow companies to create products which are associated with a higher degree of market success, but this may also reduce the scope of complements in the market by reducing novelty in products developed using such tools[4]. Finally, we contribute to the discussion of tension brought by generativity in the form of paradox of change-stability and paradox of control-autonomy (Tilson et al., 2010; Wareham et. al., 2014; Cennamo and Santalo, 2019): Provision of external boundary resources represents opening up the platform at another layer (Yoo, 2019), and as such this is another strategic openness choice on the level of platform (Boudreau, 2010). Once these tools are provided, it may increase the generativity, but may also cause overcrowding, increased imitation, and the reduction of novel complements. Our findings show that the provision of these external boundary resources may in fact strengthen both positive and negative sides of the generativity tension: on one hand it increases the mean success of products, therefore leading a more stable ecosystem, yet it also strengthens the degree to which products are similar and based around a common set of successful designs (Cennamo and Santalo, 2019).

[4] A priori, it is not possible to say whether less or more novel complements on the platform is the optimum since it could well be that majority of consumers have preferences around more conventional complements (Rietveld and Eggers, 2018), and only a small part of demand is interested in novel complements. Therefore, we are not claiming whether one or the other is optimal, but rather point out that the provision of external middleware is associated with less novelty.

2. LITERATURE REVIEW

An important aspect of strategy for platforms is devising policies to attract third-parties to develop complementary innovations, and to create conditions that incentivize them to do so (Gawer, 2014; Wareham et al., 2014). Part of this may involve providing or allowing for the existence of tools (and policies) that enable these third-parties to create products and bring them to platform more easily (Eaton et al., 2015; Ghazawneh & Henfridsson, 2013; Tiwana et al., 2010). For instance, Google and Apple provide their SDK (standard development toolkit) for software developers to create basic applications. In addition, they provide libraries that are applicable across a wide range of third-party applications, which simplify development tasks of the third-parties by allowing them to focus on more valuable software development tasks. This can be understood from the way Google describes the Jetpack suite of middleware components: *Jetpack is a suite of libraries, tools, and guidance to help developers write high-quality apps easier. These components help you follow best practices, free you from writing boilerplate code, and simplify complex tasks, so you can focus on the code you care about* (Google, 2019). However, the literature on modularity and recombinant innovation suggests that while this may simplify development tasks for developers, the implications of using middleware on the ability of developers to create products which are novel and/or valuable is not clear, also due to the process through which developers choose to use these middleware components. We review existing studies relating to these topics below.

2.1. Recombinant Innovation

Innovation is often characterized as a "recombinant process of reusing and recombining existing technologies" (Nelson & Winter, 1982) that can be traced back to Schumpeter (1942). Existing studies of recombinant innovation often focus on the production of academic knowledge, measured through the publication of academic research papers, or technological innovations, measured through patents and patent-based metrics (Garud et al., 2013; Flath et al., 2017). This also relates to the idea within information systems as innovation as a process of knowledge recombination (Alawi & Leidner, 2001).

A particular thrust of this literature has been the relationship between various measures of knowledge recombination and innovation outcomes (Fleming, 2001; Somaya, 2012; Arts and Veugelers, 2015; Schilling and Green, 2011; Katila and Ahuja, 2002; Rosenkopf and Nerkar, 2001). For instance, Fleming (2001) found that innovations based on existing components are on average are more valuable with less innovative uncertainty (i.e., less dispersion across their values). Kaplan and Vakili (2015) distinguish value from cognitive novelty, and find that using more familiar components leads to novelty, which then leads to more valuable innovations. Mukherjee et al.

(2016) have found that the most useful (or impactful) innovations are those that are based on combining conventional technological knowledge with more distant or less conventional knowledge. Arts and Fleming (2018) argue that inventions can be not novel but still be valuable, or they can be also novel but not so valuable, and in overall find that exploration of new fields are associated with higher novelty but less valuable inventions. In this study, we will follow previous studies and conceptualize novelty as a new combination of ideas or components (Fleming et al., 2007; Uzzi et al., 2013; Arts and Fleming, 2018).

However, little is known about the case where innovators can choose to build on or reuse a particular enabling technology (i.e., tool), through which the recombinant innovation process itself change. Qualitative studies have attempted to study this in specific, high impact cases (Majchrzak et al., 2004; Ben-Menahem et al., 2016). Yet, without being able to observe such enabling technology based recombinant innovation in a large number of cases, it is difficult to draw conclusions regarding the implications of these technologies for platform companies.

A body of studies has looked at the related question of how exogenous changes in the cost of using various technological components or knowledge resources change the recombinant process. For instance, Teodoridis (2017) finds that the hacking of the Microsoft Kinect, which suddenly reduced technology costs in motion-sensing research, allowed specialist researchers from other fields to enter into this research domain to work together with generalist researchers, resulting in more diverse team combinations. Similarly, Agrawal et al. (2016) found that, following the collapse of the Soviet Union, academic research in mathematics shifted as new areas of research opened up as a result of the sharing of previously unknown knowledge in mathematics that was only available in Soviet Union.

All these previous studies, however, considered the recombination of knowledge in the form of patents or academic publications, and our study, instead, looks at the decision of developers to build their innovation on top of middleware. This is an important distinction since recombination of knowledge in previous studies is non-rivalrous, whereas the use of middleware is a strategic choice. Therefore, when developers use middleware, they face a tradeoff of either using an existing technology, or creating that technology themselves from scratch. Furthermore, existing empirical studies have looked primarily at innovation outcomes such as forward citations, number of outputs in terms patents or publications, and the market-for-technology value of the innovation (Hall et al., 2005). It is implied that these results translate into product market outcomes. For example, radical innovation in terms of citations may represent radical innovations in the product market as well (Hall et al., 2005). Yet, there has not been empirical evidence linking technological recombination and reuse to product market outcomes such as revenues and product novelty (which is different from technological novelty).

This distinction between our study and previous studies can be bridged by building on the conceptualizations and findings of the literature on modularity and innovation, which complements above mentioned recombinant innovation studies as well.

2.2. Modularity and Innovation

The literature on modularity argues that firms may benefit by modularizing their innovation processes (Pil and Cohen, 2006; Baldwin and Clark, 2000; Sanchez and Mahoney, 1996; Garud and Kumaraswamy, 1995), by spliting the innovation process into interchangeable components that are connected via standardized interfaces. Existing studies have often discussed modularity in regards to how companies design their products, and the implications of this for innovation. Modularity relates to the "product architecture", which describes "the arrangement of functional elements, the mapping from functional elements to physical components, and the specification of interfaces among components" (Ulrich 1995, p. 420). Modularity (and related concept of product architecture) does not only relate to physical innovation, but are also central concepts for digital innovation and platforms in the form of layered modular architectures (Yoo et al., 2010; Gawer, 2014).

More modular innovations primarily "enable heterogeneous inputs to be recombined into a variety of heterogeneous configurations" (Schilling, 2000, p. 317) by design, therefore enabling a larger number of potential innovations at lower cost through "economies of substitution" (Garud and Kumaraswamy, 1995). A second major advantage of modularity is abstraction, which is the hiding away of the complexity underlying in components and instead building simple (or "loosely coupled") interfaces to manage the components of a complex system (Mahoney and Sanchez, 1996; Baldwin and Clark, 2000). These two main advantages enable: (i) Greater number and scope of experimentation and recombination by combining different components, and (ii) a repartitioning of the work by reducing the scope and level of knowledge required by innovators to work on the particular components of the products. These two factors in turn, potentially lead to greater product variety and performance (Takeishi & Fujimoto, 2001; Utterback, 1994; Garud and Kumaraswamy, 1995; Fleming and Sorenson, 2001). Yet, modularity may in other ways inhibit variety and performance. One main downside of modularity is that more modular products may be easier to imitate and replicate (Pil and Cohen, 2006; Baldwin and Henkel, 2015; Rivkin, 2000; Ethiraj et al., 2008; Ethiraj and Levinthal, 2004). This could then make it easier for competitors to imitate successful products, and lead to a general reduction in product variety and also has performance implications for the imitating and imitated products. Another downside of modularity is the loss of synergistic specificity (Schilling, 2000), through which a product may achieve a high performance by having custom, tightly integrated components for a given purpose. Relatedly, the use of a defined set of building blocks which also fixes some level of the design hierarchy (Clark, 1985) may create high costs for more custom applications (Arora et al., 1997)[5].

Generally, studies on modularity focus on the decision of companies to modularize their products, but focus less on the impact of using modular components on innovation outcomes[6]. Closest study in this literature for our purposes is the Fleming and Sorenson (2001), which finds that medium levels of complexity (which is the interaction of interdependence and size) is associated with higher usefulness and medium level of innovative uncertainty. However, that. study does not provide much insight into novelty[7], and is based on technological components used in individual patenting efforts, rather than the strategic choice of using a technology that enables a particular mode of recombination in innovation efforts.

Relatedly, the potential benefits of modular or "enabling" technologies have been discussed in the literature on user innovation, in particular on the innovative impact of user toolkits, which highlight similar mechanisms as the literature on modularity (Von Hippel and Katz, 2002). Namely, that companies find it difficult to cater to the wide spectrum of consumer tastes, and therefore they may allow customers to develop variations to the products themselves. This literature has highlighted the benefits that the provision of toolkits may provide by separating the required knowledge for technical design and user needs (reminiscent to benefits of abstraction in the modularity literature), fast prototyping and experimentation, and the provision of basic building blocks in the form of commonly used libraries, which in combination, lead to higher variation and better designed innovations, but has not considered the tensions that may arise due to potential limits on the novelty or performance, as described earlier above.

[5] In a way, one can also consider this issue as a particular case of loss of synergy, where the parameter of optimization is not "performance", but novelty.

[6] A subset of studies has also looked at how organizational or behavioral factors may shape impact of modularity on innovation (Lau et al, 2010; Cabigiosu and Camuffo, 2012). Another set of studies have looked at code reuse in open source software projects with the lens of modularity (Haefliger et al., 2008; Rullani and Haefliger, 2013). While these studies have looked at the reuse of lines of code, they do not consider the use of broadly applicable components such as middleware. Re-use of major components such as middleware is likely to have a major impact in innovation outcomes. In that vein, there has been a set of studies that have looked at remixing or recombining different modular components but they approach with the perspective of online communities, such as those studies on 3D printing communities (Flath et al., 2017; Kyriakou et al., 2017; Stanko, 2016).

[7] As noted earlier, and highlighted by Arts and Fleming (2018), novelty and value are not necessarily positively correlated, and it is important to separate these two dimensions rather than assuming novelty implies value.

3. THEORY AND HYPOTHESIS

3.1. Middleware and the Distinctiveness of Innovation

An important outcome of the technological recombination process is the novelty of the technology being generated (Arts and Veugelers, 2015; Arts and Fleming, 2018). The specific meaning of a novel product in this case relates to how different or unique a product is from those that have come before, rather than being a determinant of quality or value of an innovation (Castañer, 2017; Arts and Fleming, 2018). In our context, we are interested in the degree to which a product is novel from the perspective of the product features (e.g., the combination of game characters, storylines, and themes, etc. in our video games setting).

There are a number of channels through which building on a modular middleware component, which can be reused across multiple products, may be associated with products that are more novel. First, the modular nature of middleware reduces the costs of experimentation, enabling developers experiment more easily and arrive at more novel products (Garud & Kumaraswamy, 1995). This is similar to what earlier studies of recombinant innovation have found, when looking how the recombination of knowledge leads to the creation of innovative outputs (Kaplan and Vakili, 2015; Arts and Veugelers, 2015; Fleming, 2001). Second, since using pre-existing, modular, middleware components reduce the costs of creating new products, software developers can reallocate the resources they would have had to spend on creating basic functionality (now done through middleware), to more creative tasks that could distinguish their products from others (von Hippel and Katz, 2002; Arora et al., 1997). This could lead to the development of more creative and distinct products. Additionally, given that software development is generally carried out in teams (of designers, programmers, etc.), this could reduce the need for technical expertise, enabling individuals with more diverse and less technical backgrounds to contribute to these projects. This increase in diversity could then lead to more novel outcomes (Singh and Fleming, 2010; Teodoridis, 2017).

However, there also exists a mechanism through which the use of these modular components may limit the creation of game titles which are more novel. As much as it is a shift towards modularization and a reduction of costs, middleware also constitutes a decrease in the flexibility of how certain products can be developed (Fleming and Sorenson, 2001). It becomes difficult to customize features that are built into the middleware components, and to adapt them beyond what the middleware allows, which can be explained by "fixation" of higher level design hierarchies through the adoption of middleware (Clark, 1985). For instance, licensed game engines, which constitute the common form of middleware in video game development, allow game developers to simplify much of the tasks in development. However, if a game engine does not contain some features (e.g., certain game engines for mobile phones do not allow you to show 3D objects), then it becomes very difficult or costly to customize the existing set of features provided by the tool. Therefore, developers have a set "feature space" which gets exploited over time, making it difficult to create products which are different from those in the marketplace. This is consistent with the idea that, with the general purpose technologies that can be used as the building block of innovation, customization costs can quickly add up to prevent using such technologies if the intended application domain of the final innovation is distant than the proximate application domain of the general purpose technology (Arora et al., 1997; Bresnahan and Gambardella, 1998; Gambardella and Giarratana, 2013). Therefore, while these modular technologies may reduce the cost of creating innovations, using them incurs customization costs if the middleware is used for more novel purposes. Finally, these negative effects on novelty might be further compounded by competitive interactions among different developers in the marketplaces. Building on a modular structure may make it easier for competitors to replicate successful product designs (Ethiraj et al., 2008; Baldwin & Henkel, 2015), which could lead products which use middleware components to be even more similar.

Since it is unclear, a priori, which one of these effects is dominant, we hypothesize the above arguments on the impact of middleware use on the product novelty as competing hypotheses:

Hyp 1a. *Middleware use is associated with products which are* ***more*** *novel.*

Hyp 1b. *Middleware use is associated with products which are* ***less*** *novel.*

3.2. Middleware and Product Demand

Perhaps the most important outcome regarding the use of modular components for innovation is how this influences the usefulness, or market value of a particular technology (Fleming, 2001; Majchrzak et al., 2004). Existing studies have looked at this from the perspective of patents, and knowledge recombination. Yet, as mentioned in earlier sections of this paper, we do not understand how this affects product market outcomes, such as usefulness in terms of observed product market demand for products. This can be thought of empirically as the total sales of a product, within the platform it is being sold on.

There are a number of different arguments for how middleware may be associated with positive market outcomes, in the form of greater demand or more revenue. Similar to the arguments regarding the impact of middleware on the novelty of products created, building on modular middleware components may allow companies to shift resources away from mundane and repeatable development tasks, and focus efforts on creative or value enhancing activities, leading to more valuable products. Additionally, the modularity allows developers to experiment more easily, allowing them to experiment until they arrive at the most valuable version of their products (Fleming and Sorenson, 2001; von Hippel and Katz, 2002).

However, the use of these modular components may also be associated with products which are less commercially valuable. On the one hand, the use of middleware components may allow companies to reduce the costs of development, potentially leading to a "race to the bottom" as less professional entrants begin to join the marketplace. Boudreau (2018) demonstrates this phenomenon in the case of mobile apps developers. There is also anecdotal evidence from Boudreau and Hagiu (2009) regarding this phenomenon in the Atari third party marketplace, where the low cost of development lead to lower quality products flooding the market. Additionally, given that modularity may make it easier for competitors to replicate successful innovations (Baldwin & Henkel, 2015; Ethiraj et al., 2008), we could expect developers to replicate successful products, reducing the expected demand for any one product. Finally, the use of modular components may restrict achieving synergistic specificity (Schilling, 2000, Tiwana, 2018), through which increased value is drive by idiosyncratic modules within an innovation. Therefore, it is possible that there exists an association between the use of middleware and less valuable, or lower revenue generating products because of the development of lower quality products.

We frame these countervailing arguments as competing hypotheses, which we attempt to resolve in our empirical analysis, as it is unclear a-priori which of these effects would dominate.

Hyp 2a. *Middleware is associated with* ***more*** *commercially valuable products.*

Hyp 2b. *Middleware is associated with* ***less*** *commercially valuable products.*

4. DATA AND VARIABLES

We study the use of middleware and subsequent product outcomes in the console gaming industry between 2001 and 2010, during the time that Generation 6 and 7 of video game consoles were available in the marketplace. We focus specifically on the use of licensed (3rd party) middleware by game developers in developing complements for these period's game consoles. We chose this period, because in 2001, licensed middleware was officially introduced into the console game industry (through Sony's Tool & Middleware Program for Playstation 2, which is followed by other platform owners shortly after), allowing us to study a period where there was heterogeneity in the use of middleware (which is now widespread).

We assembled data from multiple sources to perform this analysis. We collected data on releases of console games from Moby Games, which contains the most detailed historical information about the games available on all major video gaming consoles. This data has been used in earlier studies (Mollick, 2012; de Vaan et al., 2015; Cennamo et al., 2018), specifically because of its rich history of the video game industry. This data includes information about quality ratings, as well as detailed information about the composition of the developer team. These data also include rich product descriptions, describing the content of the game. Finally, this data also contains information about game engines, physics engines, graphics/3D engines, as well as other middleware tools (such as those used for animation, or creating environments to use in games) which are the most common forms of middleware used. This also reflects whether the middleware was licensed by a third-party to the developer – a critical issue as we focus on such licensed middleware in our analysis. In addition, this data was combined with detailed sales information from NPD that includes the demand (revenue) generated by every title available, separately each specific console release of that title (hence, providing revenue data at the title-platform level). Our final data consists of 1112 title-platforms released for Sony and Microsoft owned platforms (Playstation 2, Playstation 3, Xbox, and Xbox 360) within genres that the adoption of middleware components was observed[8].

4.1. Measure of Middleware

Our middleware measure intends to capture whether firms are using middleware, licensed by a third party (other than those provided by the platform owner or inhouse technologies owned by the game developer) that can be used across multiple products (i.e., games). These middleware components broadly fall into game engines, 3D engines, graphics engines, physics engines, and other tools (this may include smaller tools for animation, environment building, networking etc.). Examples of middleware components includes well known 3rd party middleware components such as the Havok physics engine, or the Unreal game engine[9]. We look both at the total number of middleware components used in each product, as well as dummies for individual types of middleware to test the robustness of the effects. Middleware components are also designed to be combined multiples of them modularly – with the most common combination being the use of physics engine with a game or 3D engine as they tend to cover different aspects of game

[8] We began with the entire population of Moby games data, but then reduced our sample in order to make the most meaningful comparison. We began by limiting ourselves to the time frame between 2001 and 2010 when middleware was beginning to be introduced. We focused only on console games for Playstation and Xbox which were the two dominant platforms and similar in terms of the games available. Middleware first emerged on the PS2 platform, and so this is where we focused our empirical analysis. Middleware components also existed only in certain games (action, racing, etc.). We therefore limited the sample to only categories where middleware was at some point introduced.

[9] Unity, which is currently the most widely used game engine was not available for these platforms in these generations. The closet analog is the Unreal game engine, which was the original basis for many of these games.

development. The total number of middleware components reflects the extent to which the code within the game is being created using middleware, or being hand coded by developers.

4.2. Measure of Performance

We measure the commercial value of individual title-platforms (e.g., FIFA 2005 for Xbox360) based on the revenues that the title-platform generated in the U.S. Console Game Market during its lifetime (on average, 80% of revenues for a title-platform is generated in its first 12 months). NPD research data provides us with monthly sales of games in U.S. market, for each platform, from January 1995 until February 2010. Since our sample ends in 2010, our sales data is not truncated for products that were released later, since we are able to observe the entire period that games are on the market, including those released in 2010[10].

4.3. Measure of Product Novelty

As described thus far, our notion of product novelty relates to the concept of how different a product is from those that have been previously released. The inherent challenge in measuring the degree of novelty is defining whether something is different from the existing set of products, particularly when all product development inherently involves variation and doing things that are somehow novel. We construct our measure of novelty on the basis of the Moby Games text description of each game, that captures succinctly the objective, ideas and concept of a particular game[11]. Importantly, the description does not contain information about middleware used and other technical components. A common approach in determining novelty from text data involves constructing measures based on angular distance (cosine distance or Euclidian distance) between vectors based on product descriptions. However, these measures have limitations, because they do not account for the fact that all products are to some extent "differentiated" from existing products (however, we include them as robustness checks). Therefore, existing studies have not been able to qualitatively distinguish products that are in fact novel.

We adopt techniques from anomaly detection, a commonly used machine learning approach, to construct a measure of novelty. Most classification problems in machine learning require pre-

[10] Note that we will be controlling for "release year" fixed effects in our regressions, effectively controlling for differences in the time on market for each product – therefore games released towards the end of our observation window will have been "corrected" by this fixed effect.

[11] Existing studies have measured this on the basis of "tags" or labels from Moby games. This provides a coarser description of a particular game, and is subject to the bias in describing or categorizing the games in relation to other games. The descriptions used in the present analysis are provided by the game developer and describe the features of the game. We find our main results are unchanged using such "tags", as presented in Appendix Table D1. However, the use of these "tags" has little variation after controlling for category (i.e., genre) fixed-effects, therefore not informing on fine-grained differences in novelty across titles, but also overstates differences if there is only one tag between otherwise similar games is different. In sum, this further justifies our method to measure novelty, but we are also able to show our results hold with the previously used measure.

specifying groups (i.e. A and B) and training an algorithm to distinguish between different observations (characterized by a vector of characteristics). However, there are cases where it's difficult to specify these groups prior to training a model. For example, machine learning techniques are often used in data security or manufacturing to identify when future data are different from past data. An application of this is to identify problems in manufacturing when performance or quality metrics in the future differ from these metrics in the past. Anomaly detection is a classification technique developed for such instances, where there is considerable data on current situations, but the objective is to model whether future data differs from these earlier data. This technique works by fitting a contour (surface) around the training data (data from the current period, time *t*) and then checking the position of the test data (from the following period, time *t+1*) relative to this contour. Observations within the contour can be thought of as describing the area covered by existing data and therefore can be thought of as being similar to existing data (regular or normal) while observations outside of the contour can be through of as being different (irregular, anomalies or abnormal) observations.

We adapt this to measuring the novelty of individual software titles based on their software descriptions. Using the software descriptions, we removed punctuation, tokenized the data (removed grammar and suffixes) and converted the descriptions into term frequency vectors (a vector for each title, with k terms for the frequency of each word that occurred in the description. This approach ignores word order, meaning, sentiment and more complex context. It is what is referred to as a "bag of words" approach. Once we converted the text data to numerical (term frequency vectors) we can proceed with the novelty detection algorithm.

For each year of the data we construct two samples. We split our numerical representation of the text data (term frequency matrix) for all titles prior to the year in question and the sample (term frequency matrix) for the focal year. We then run our anomaly detection algorithm to fit a contour around the sample from previous years. This contour can be interpreted as the "novelty frontier" which defines the boundary for the numerical representation of the data. This was done using a one-class support vector machine, with an RBF kernel[12]. Observations outside of the novelty frontier are distinct or different, from those released earlier. Observations within the novelty frontier (which can be thought of as a circular or elliptical boundary as seen in **Figure 1**), are less distinct and more similar to those which were released earlier. This approach also provides us with a measure of how far each observation is from the novelty frontier, which reflects a more granular measure of the degree of novelty.

[12] RBF (Radial Base Function) Kernel is the most common approach used for one-class SVM machines. A maximum misclassification rate of 0.5% was allowed. However, the method was robust to alternative specifications. See Appendix A and C.

We tested the robustness of our results to conventional measures of distance, such as angular distance between the term frequency matrices described above. We also tested topic modeling and other dimensionality reduction techniques and computed the metrics using these refined data to test the robustness of the results (see Appendix **C**). Finally, we also tested our results on the basis of "tags" or labels from Moby games (see Appendix **D**), the coarser measure as used in previous studies (de Vaan et al., 2015; Zhao et al., 2018).

-----------------------------------Insert Figure 1 about here -----------------------------------.

4.4. Additional Controls

In our analysis, we include a number of control variables to capture different factors that may be influencing the results. In all of our regression analysis, we include multiple fixed effects at the: Year (release year of game), firm (publisher) and category (product market niche/genre) levels. We also included a number of time varying controls, to absorb potential confounding factors that may not be captured by the fixed effects. We include as a control variable, *Product Experience* which is the log-transformed number of products that the firm has previously released. We also include as a control *Middleware Experience*, which is a log-transformed count of the number of products previously released that used licensed middleware. Including both of these variables simultaneously helps to capture the size of the firm and its overall product portfolio, as well as the proportion of those titles that were based on middleware.

We also include *Licensed Title* which is an indicator variable, equal to one if the rights to make the game were licensed from an outside entity. An example of this would be James Bond based game 007: Goldeneye, which was built around the James Bond IP. This may shape both the novelty of the game and demand, and therefore is included as a control. *Inhouse Middleware* is an indicator variable that indicates whether the firms that are using middleware components that they developed themselves. As a reminder, our measure of middleware relates to third-party middleware components developed by outside companies. *Project Size* is a log-transformed count of the number of technical credits (programmers, designers, artists, and engineers) involved in creating the game. This is a common proxy for the budget of the actual game, since the primary input for game development is human capital such as programmers and game designers. This measure has been used in a number of other papers (Mollick, 2012; de Vaan et al., 2015).

5. ANALYSIS AND RESULTS

We begin our analysis by providing descriptive evidence for how the use of middleware maps to patterns of product novelty and demand. We affirm the robustness of these descriptive patterns using regression analysis in later sections.

5.1. Descriptive Evidence on Product Novelty

As a first attempt in understanding the patterns between product distinctiveness and middleware use, we compare the share of titles that were novel (i.e. beyond the novelty frontier) by different levels of middleware in **Figure 2**. Here we see that products which used fewer middleware components were more novel, while those that used a greater number of middleware components were less novel. In the extreme case where products were based on three middleware components, there were no titles that were beyond the novelty frontier. This provides the first evidence in support of *Hypothesis 1b.*

Next, we compare the distribution of novelty scores (our measure of novelty calculated as described in earlier sections) stratified by the use of middleware[13]. We present the stratification based both on the number of products. In **Figure 3**, we present gaussian kernel density estimate for the distribution of novelty scores for products with and without middleware. We can see that products with one middleware component have a distribution that is shifted slightly to the left of those that do not use any middleware components, suggesting that those which do not use middleware are more distinct. Finally, those that use two middleware components are shifted even further to the left suggesting that they are even less distinct novel. These results further suggest that products which use middleware components (middleware) are on average less distinct than products that do not use them, consistent with *Hypothesis 1b.*

5.2. Descriptive Evidence on Product Demand

Here we explore descriptively the relationship between the use of middleware and the demand for a particular product, measured by the total revenue that it generates, on a specific platform. We again begin in **Figure 2** by comparing the total sales (in USD) of each title-platform, based on the number of middleware components observed[14]. We find that revenues are higher in instances where middleware is used, suggesting support for *Hypothesis 2a.* We then compare the distribution

[13] As described in earlier sections, negative novelty scores indicate that a product is beyond the 'novelty frontier' (products that are novel), while positive scores indicate a distance away from the novelty frontier (products that are not novel).

[14] All measures of product revenues are based on the revenues generated on a particular platform (i.e., title-platform level). Therefore, we alleviate issues that are brought by the potential economies of scope across platforms at the title level (which would be the sum of title-platform sales across all platforms of release) with the use of middleware (Corts and Lederman, 2009).

of our outcome variable, total product revenue for the title-platform, using a gaussian kernel density estimate for products that use middleware and those that do not, in **Figure 3**. The revenue measure here is corrected for year, niche (genre), publisher and product size. We find that the distribution is shifted to the right for those titles that use middleware, and further more so for those that use two middleware. The means of the distributions appear to be slightly shifted. However, the results are not as drastic as those for product distinctiveness. While this provides some support for *Hypothesis 2a*, we move to regression analysis in subsequent sections, where we can test for the statistical significance of these differences.

-----------------------------------Insert Figures 2 and 3 about here ----------------------------------

5.3. Regression Analysis for Product Novelty

Here, we look for a more stringent test of whether the use of middleware is associated with lower product novelty consistent with *Hypothesis 1b* and the earlier descriptive evidence. Given that our outcome variable is normally distributed (see **Figure 3**), we use an OLS regression with the novelty score as our outcome variable[15]. We present the results of these regressions in **Table 1**. In columns 1 through 3, we introduce our measure of middleware use along with time dummies, publisher (firm) and category fixed effects, as well as additional control variables. The results suggest that middleware use is associated with less novel products and the results are significant at the 1% level. In columns 4 and 5, we split the middleware variable into different components including individual dummies for the number of middleware components that are used (e.g., 2 or 3 middleware components), and dummies for the individual components (e.g., game engine, 3D engines, etc.). The results in column 4 suggest that more middleware is associated with lower distinctiveness, consistent with earlier results, while the results in column 5, suggest that different components may contribute differently, but that individual components constitute a much weaker (or noisier) effect than the combination of these components. These results provide support for *Hypothesis 1b.*

Regarding the magnitude of these effects, one additional middleware component corresponds to a decrease in the novelty score of 0.003. This is a within group effect, that can be interpreted as the change in the novelty score of a developer within a particular category, in a given year. The standard deviation, within group, in novelty is approximately 0.005. Therefore, a shift

[15] This provides a more reliable analysis than a dummy variable for whether a particular product is beyond the novelty frontier (i.e., Novelty = 1 dummy), because there is some subjectivity in the algorithm in terms of where the frontier is located. This depends on parameter choices and the kernel function for the SVM classifier. The novelty score therefore provides a more reliable measure since the relative novelty scores of less versus more novel products will not vary greatly under different parameters or classifier kernels.

from zero to two middleware correspondents responds to a decrease in the novelty score of more than one standard deviation. We provide some examples of games together with their descriptions and novelty scores to illustrate the impact of these novelty score changes in Appendix.

Finally, we provide a number of robustness checks in the appendix, including a series of robustness checks with more conventional measures, such as cosine distance of text descriptions, or product tags, which are not extracted from the text descriptions, but generated by the publishers / users to classify the game title (**Appendix D**). The results were also robust to various parameter choices in setting up the one-class SVM algorithm (in **Appendix A and C**).

-----------------------------------Insert Table 1 about here -----------------------------------.

5.4. Regression Analysis for Product Demand

Here, we look for a more stringent test of the impact of middleware components on product demand measured by product sales, on each platform. We use an OLS regression with the log-transformed product (title-platform) revenues as the outcome variable for our analysis. We present the regression results in **Table 3**. In columns 1 through 4, we include our main variables of interest and controls including year, publisher and genre fixed effects as well as our set of controls from the earlier regression. The coefficient for *N. of Middleware Components* is positive and significant at the 1% level, suggesting that middleware use is associated with more valuable products. In columns 5 and 6, we split our measure of middleware into the number of middleware components (column 5) and into the types of components (column 6). The results suggest that using a higher number of middleware components is associated with higher demand, but simultaneously using individual components does not appear to be related to higher demand, as indicated by the negative but not significant coefficients in column 6. In column 7, we introduce the novelty measure from **Table 1** as an additional control variable. The novelty coefficient is insignificant, while the baseline effect for middleware use remains consistent in sign and significance. These results provide support for *Hypothesis 2a.* In terms of the magnitude of the coefficients (column 4) correspond to 29% higher revenues on average, with each middleware component used[16]. While this magnitude is relatively large, the coefficient is 18% of the within group standard deviation (with developer, industry and time period), suggesting a high degree of variability in revenue, even within developer firms and products.

-----------------------------------Insert Table 2 about here -----------------------------------

[16] Since the outcome variable is log transformed, we calculate this ratio based on the exponentiated function $\exp(\overline{x} + \beta)/\exp(\overline{x})$.

5.5. Robustness Checks and Supplementary Analysis

The results thus far support *Hypothesis 1b and 2a.* The theoretical arguments leading up to these predictions both relate to the technology itself and the companies that choose to use this technology. To further explore these relationships, we look at the decision of which companies choose to adopt middleware components.

For instance, the main rationale behind middleware is that it provides a modular base that can be reused and built upon across multiple products. These middleware components are meant to replicate the basic functionality that developers would have to create for each successive title, and therefore can lower the costs of development. However, as we argue in the theory development, this may also lead companies to create less valuable products since the costs of development are lower. Or, it might lead companies to reallocate their resources to create more valuable products. The patterns we are observing in support of *Hypothesis 1b* and *2a* suggest that middleware components are associated with products that are more valuable but less novel. This runs opposite to the argument that middleware increases "low cost" imitation, as it has been observed in other settings (e.g., smartphone apps, Boudreau, 2019). Rather, it suggests that middleware components allow companies to reallocate their resources to create more valuable products, but that the technology itself makes it more difficult to create products which are more novel from those already in the marketplace.

5.5.1 Developer Characteristics and Middleware Use

To explore the discussion above directly, we estimate the likelihood of developers using middleware. The main variables of interest are *Project Size* and whether a developer is an entrant (i.e., if this is developer's first product release). In **Table 3**, we find a positive relationship between being an entrant and the number of middleware components (column 1 & 2), as well as a negative relationship between the number of product releases (column 3 & 4), the inverse of being an entrant. In columns 5 & 6, we split the sample between entrants and firms that have previously released at least a single title. We find that project size is associated positively with middleware use for all of the columns, except column 6, in which it is insignificant. Therefore, even in the case of entrants (column 6) we do not find that smaller project size is associated with middleware use. This suggests that while middleware may replace the functions of many human workers, it does not lead to "low cost" development, but instead companies that use middleware build on top of this technology. This is consistent with the view that middleware components do allow for the creation of more reliable and valuable technologies, but this may limit the ability of companies to distinguish themselves through more novel products.

----------------------------------Insert Table 3 about here ----------------------------------.

5.5.2 Instrumenting for Factors Leading to Middleware Use

As a secondary check, we attempt to account for factors that may influence the use of middleware and check whether our findings remain stable or change. This would indicate whether our outcomes are associated due to the middleware's technological features, or whether our outcomes are driven due to which companies choose to use middleware. A central argument how middleware affects product development is that, the use of middleware lowers software development costs, enabling companies to shift resources from having programmers creating basic functionality for the game to having a modular component that can be re-used across multiple games. This enables companies to hire fewer programmers to develop basic core technologies, but instead focus on other development tasks, such as game testing, design or other creative tasks. We exploit a policy change in the way H1-B Visas were issued, that affected the ability of companies to hire and retain their workers, but not product market outcomes such as novelty or product market outcomes[17].

Console game developers, like many other technology and software firms, are largely US-based and rely heavily on foreign talent often through H1-B visas. In 2004, the US imposed a quota on H1-B visas that made it more difficult to acquire this foreign talent. U.S. software developers that wanted to acquire talent found it more difficult to do so after this shock. However, those that already had previously hired workers on H1-B visas had an even more captive, and in turn cheaper, workforce because these workers were not able to switch to other companies (because they would have to get a new H1-B visa). We collected data on the applications and grants of H1-B visas for all US firms, and matched these to the firms that existed in our sample[18]. This provided us with information about which firms had relied on software engineers and game developers from foreign countries through the H1-B program.

We construct two instruments: 1) A count of the number of employees that were previously hired through the H1-B program (U.S. developer firms only, including U.S. subsidiaries of foreign parent firms), and 2) an interaction of the number of employees previously hired under H1-B, with an indicator for the period after the policy change. This captures the effect of the policy change on those companies that had a larger base of foreign employees. The indicator of the period after the shock (the missing term in the interaction) is omitted from the analysis because it is captured by the year dummies in the main regression. Fixed effects are included at the firm, product category and year level to capture other potential temporal differences in the application of H1-B visas, as well as differences between companies that use H1B visas and those that do not. Also, in the

[17] Raffiee et al (2019) study the H1B program, and find that workers on H1-B visas have more restricted mobility, consistent with these arguments.

[18] This information is provided by the U.S. government as a public record, https://www.foreignlaborcert.doleta.gov.

Appendix, we provide evidence that H1B Policy requests decline following the policy event, and that the majority of H1B visas for companies in the sample were for programmers, which are those directly affected by middleware use.

In **Table 4**, we present the 2SLS instrumental variable results for both novelty and revenue outcomes. In the bottom part of the table, we present the results of the first stage regressions. The outcome for the first stage regression is the *N. of Middleware Components* that has been used in the analysis as the main explanatory variable thus far[19]. For our first stage instruments, we find that those firms which a greater number of H1B visas are more likely to use middleware components (approximately 4% higher per 10 H1B employees), but that after the policy change the effect is reversed, and show a 5% decline in middleware use per ten H1B workers in the likelihood of using middleware components. These coefficients are significant at the 5% level, and the first stage multivariate F test is also significant at the 1% level. These results are economically significant as companies in the sample have as many as 150 workers with H1B visas, and having a substantial share of workers with mobility constraints may greatly shape software use decisions. These results are consistent with the process described above.

The second stage regression results are consistent with our earlier analysis, in that we find a significant (at $p < 0.1$) and negative relationship between middleware use and product novelty, while we find a positive and significant relationship between product demand and middleware (at $p < 0.1$). These results suggest that middleware components are associated with products that are less novel, while being associated with products that are more valuable. While the results from the previous **Table 3**, suggested that middleware components are used with specific kind of firms (new entrants) or projects (large sized projects of incumbent firms), even after accounting for these with our 2SLS model, we still find that middleware components are associated with more popular titles, but products which are less novel. This suggests that the middleware technology plays a role in leading to the creation of products which are more valuable, but which are less novel or distinct.

-----------------------------------Insert Table 4 about here----------------------------------.

5.5.3 Specialization in Middleware Based Development

While the earlier results suggest that technology itself plays an important role, in shaping the types of products that are created, the choice firms have in using middleware may also play an

[19] As first stage of the 2SLS model is a count variable, using ordinary 2SLS coefficients may provide a poor second stage prediction (as it is based on a normally distributed prediction). We follow the approach suggested by Angrist and Pischke (2009, Section 4.6.2, Forbidden Regressions) as an alternative to the forbidden regression of substituting the non-linear fitted values as instruments. While our chosen alternative is not a commonly known approach, we validated this by checking other sources. See Wooldridge: https://www.statalist.org/forums/forum/general-stata-discussion/general/1371733-how-to-handle-endogenous.count-variable-in-impact-study We report the first stage coefficients for predicting the outcome variable, rather than the fitted value, as this helps reflect the validity of the instruments.

important role. More specifically, the use of middleware may not be a one-off decision, but firms may become repeated users of middleware and specialize in middleware-based development. While we control for this in our earlier regression results models, here we focus on the specific impact that experience with middleware development may have on development.

By using middleware on one project, developers are forgoing the option of developing basic underlying technological components and instead focusing on customizing that middleware for their specific application. In doing so, they will become more efficient at using middleware to create products, as well as developing assets, skills and human capital in middleware-based development. For subsequent projects, if they were to return developing without using middleware, they would have to abandon any middleware specific assets and instead reinvest in developing the basic development skills that were previously being performed using middleware. In this case, developers would be more and more likely to specialize in middleware-based development, and in turn developing middleware specific skills, assets and expertise.

This specialization in middleware-based development would also imply that the most experienced or specialized firms would be adept at the process of experimenting with middleware, allowing them to arrive at the most valuable combinations more easily (Kogut and Zander, 1992; Cohen and Levinthal, 1990; Kim and Kogut, 1996). Additionally, if the use of middleware allowed firms to reallocate resources to more creative or more value creating tasks, then we could expect that those specialized in middleware would be able to extract the most value out of middleware when creating middleware-based applications as they have organized themselves around these technologies.

At the same time, the focus on building and recombining a core set of technological components may have its limits. As Kim and Kogut (1996) suggest "The repeated application of a particular set of technologies or organizing principles eventually exhausts the set of potential combinations" (p. 285). This implies that the specialization in middleware development, while efficient and able to create value, may make it more difficult to create novel or distinct technological components. This is likely to increase as the degree to which these components are reused increases, and the potential set of combinations is increasingly exhausted. This in turn implies that the specialization in middleware-based development is likely to be associated with lower novelty.

Table 5 shows results regarding the specialization in middleware-based development affects our results. Both for novelty and demand, we see that the interaction *Middleware Experience x N. of Middleware Components* subsumes the previously negative (for novelty) and positive (for demand) effects of the previously *N. of Middleware Components* main variable. This finding suggests that when firms begin to initially use middleware, they are not able to generate higher commercial revenues or develop less novel products. However, as they become more experienced in using middleware, they become more specialized, experiencing higher customization costs and less willingness to develop basic functionality. This logic is consistent with the novelty and revenue results we present above.

-----------------------------------Insert Table 5 about here ----------------------------------.

6. DISCUSSION AND CONCLUSION

In this paper we studied how external boundary resources relate to novelty and commercial success of products created. We analyzed videogame titles released for Sony and Microsoft owned consoles between 2000 to 2009, a nine year period around the official introduction of the 3rd party middleware tools to gaming consoles, and found that complements using these external boundary resources were associated with less novelty, but with more commercial success. This finding has important implications from the perspective of platform strategy, as platform companies devising policies regarding 3rd party tool availability and use must consider its consequences for the platform, and its generativity, in particular. In addition, our findings have broader implcations for studies on modularity and recombinant innovation. In subsequent sections we discuss the theoretical and policy implications of these results.

6.1. Contributions to Theory

Studies of platforms have increasingly considered the key role of boundary resources, in that they serve an important role in fostering a platform's generativity by affecting the availability of complements on a platform, as well as control mechanisms via policy and norm setting (Eaton et al, 2015; Ghazawneh and Henderson, 2013; Karhu et al., 2018). As such boundary resources are closely related in reducing the unwatented variance, such as fluctuations in quality across complements in a platform (Cennamo and Santalo, 2019) and increasing the desired variance, such as variety of complements (Boudreau, 2012; Wareham et al., 2014). However, studies so far focused on boundary resources provided by the platform, and have seldom addressed the layer of independent companies that specialize in the development of 3rd party boundary resources in the form of development tools or enabling technologies. We join to the list of few recent studies (Yoo, 2019; Kang et al., 2019) in studying the impact of these boundary resources, and uniquely focus on the novelty and commercial success dimensions of complements to explain its platform level implications from these dimensions.

Although boundary resources foster innovation, and therefore impact complementor incentives to innovate (Gawer, 2014), focusing solely on complement numbers precludes multiple dimensions of governance and complement outcomes that is essential to understand the platform evolution and how it matches with the changing demand and complementor patterns while keeping its core (mostly) intact (Wareham et al., 2014; Cennamo and Santalo, 2019). Our finding that the use of middleware relates to reduced novelty, but higher commercial success of complements directly relates to this issue: Boundary resources in this case do not necessarily foster the variety of complements, but instead allow complementors to focus their efforts on a narrower set of complements that is more likely to have a higher appeal to users. On one hand, this result shows that external boundary resources may reduce undersired variance in the form product outcomes (e.g., quality), but also may end up reducing desired variance, as such technologies may put constraints to the types of complements to be developed due to various factors we discussed above. Perhaps most interestingly, we would have expected both more novelty as well as more commercial success, due to one of the key mechanisms for the middleware through the reallocation of resources from programmers to the other creative roles (or other programming roles). Although for product success, this mechanism seems to have driven positive results, for the novelty, it is likely that the customization costs due to limits of the middleware have been stronger – as exemplified in the following anectode from a classic video game, Deus Ex, in which the game director summarizes the resource allocation, customization, and also the middleware specialization issues succinctly with their experience of licensing Unreal Engine:

> "We went into Deus Ex hoping that licensing an engine would allow us to focus on content generation and gameplay. For the most part, that proved to be the case… The dollars and cents of the deal were right, and I didn't have to hire an army of programmers to create an engine… [W]e were able to make what I hope is a state-of-the-art RPG-action-adventure-sim with only three slightly overworked programmers, which allowed us to carry larger design and art staffs than usual… Technology forced design changes, too. It took time to become familiar with the Unreal engine… There were times when we should have ripped out certain parts of the Unreal Tournament code and started from scratch (AI, pathfinding, and sound propagation, for example). Instead, we built on the existing systems, on a base that was designed for an entirely different kind of game from what we were making… I guess the fact that we'll be licensing technology for our next round of projects, Deus Ex 2 and Thief 3, says the price was right. But it

> remains an interesting dilemma, and we will be able to approach our next licensed engine with the wisdom gleaned from using Unreal for this project."[20]

In addition to our main contribution above, we also contribute on the literatures on modularity and recombinant innovation. Existing studies have long considered how the modularity of technological components may have implications for innovation, both from the perspective of being able to experiment and generate novel innovations (Ethiraj and Levinthal, 2004), as well as the challenges of being able to prevent imitation (Ethiraj et al., 2008; Baldwin and Henkel, 2015). However, the use of modular technologies in product development decisions may also coincide with the decisions of how companies allocate workers, which may influence the types of products generated from these modular technologies. However, the literature on modularity has not generally focused on this issue.

Finally, studies from recombinant innovation have looked at how the use of certain technologies shapes the novelty and usefulness of technologies (Fleming, 2001; Fleming and Sorrenson, 2002). However, these studies have generally focused on technological recombination of knowledge, measured through patents. This is different from the use of tools or modular components, as patent citations (used by Fleming, 2001; Fleming and Sorenson, 2002) refer to related areas of technological innovation. However, innovators do not cannot recreate the technology in the patent if they do not which to license it (this is the purpose of patent protection). In the case of modular technologies, developers can re-create the technology themselves if they do not wish to use the modular components. This paper provides evidence of how this may work in the context of modular components, as described above.

A limitation of our study is that we were not able to focus on the time-based evolutionary dynamics of the platfoms in consideration, and how the impact of external boundary resources on complement novelty and commercial success changes across time within each platform (which we rather only control for with yearly fixed effects). This is an important further research avenue to better understand the role external boundary resource in platform evolution given our findings. Eggers and Rietveld (2018) highlighted that due to demand heterogeneity in the adoption patterns of platforms, later adopters prefer products that are less novel and more commercially successful – a finding that resonates with the impact of middleware in our study. On the other hand, Cennamo and Santalo (2019) found that part of the complementors that join in the mature stages of a platform freeride on the success of earlier complements and their contribution to the platform success, which would be incompatible with our findings. It would be not surprising to see both of

[20] Spector, W., "Postmortem: Ion Storm's Deus Ex", https://www.gamasutra.com/view/feature/131523/postmortem_ion_storms_deus_ex.php, accessed 30 January 2020.

these evolutionary effects would interact with the use of external boundary resources dynamically across platform evolution.

6.2. Contributions to Practice

From the perspective of a platform owner, or an entity that wants to offer boundary resources such as middleware to enable innovation, the results of this study provide important insights into how this may end up shaping innovation outcomes. The important conclusion is that middleware components have an important tradeoff at the core. They both enable innovation and lead to the creation of more valuable innovations, because they allow firms to avoid having to recreate basic functionality. However, at the same time, they impose constraints that limit the ability to create more novel components, and in turn creativity. Simply looking at this does not provide the full picture. Given the strong selection mechanisms at play, as demonstrated by both the theory and the analysis, the availability of middleware may greatly shape the types of products being created. For instance, middleware does not on its own limit creativity. However, it does create customization costs in relative terms and therefore shifts product development towards less creative and less novel products. Similarly, while middleware may make it more difficult to differentiate a product from others, it allows firms to reallocate their resources towards more valuable development tasks leading to more valuable innovations overall. Therefore, choosing how, whether and when to offer these boundary resources such as middleware components directly shapes the direction of the innovation process that occurs. If this is in the hands of a platform owner, then introducing middleware comes with a clear tradeoff and one that has to be carefully managed in order to optimize the level of innovation that is occurring.

REFERENCES

Agrawal A., Goldfarb A., & Teodoridis F. (2016) "Understanding the Changing Structure of Scientific Inquiry." American Economic Journal: Applied Economics, 8(1): 100-128.

Alavi, M., and Leidner, D. E. (2001). "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," MIS Quarterly, 25(1), pp. 107-136.

Arora, A., A. Gambardella and E. Rullani (1997) 'Division of Labour and the Locus of Inventive Activity,' Journal of Management and Governance, 1, 1–19.

Arts, S., and Fleming, L. (2018) Paradise of Novelty—Or Loss of Human Capital? Exploring New Fields and Inventive Output. Organization Science, 29:6, 1074-1092.

Arts, S., and Veugelers, R. (2015). Technology Familiarity, Recombinant Novelty, and Breakthrough Invention, Industrial and Corporate Change 24(6): 1215–1246.

Baldwin, C.Y., Clark, K.B., (2000). Design rules. The Power of Modularity, 1. MIT Press, Cambridge, MA, USA.

Baldwin, C. Y. (2008). Where do transactions come from? Modularity, transactions, and the boundaries of firms. Industrial and Corporate Change, 17(1), 155–195.

Baldwin, C. Y. and Henkel, J. (2015), Modularity and intellectual property protection. Strategic Management Journal, 36: 1637.1655.

Ben-Menahem, S., von Krogh, G., Erden, Z., & Schneider, A. (2016). Coordinating knowledge creation in multi-disciplinary teams: Evidence from early-stage drug discovery. Academy of Management Journal, 59(4): 1308–1338.

Boudreau K, Hagiu A. (2009). Platforms rules: multi-sided platforms as regulators. In Platforms, Markets and Innovation, Gawer A (ed). Edward Elgar: London, UK: 163–191.

Boudreau K. (2010). Open platform strategies and innova-tion: granting access vs. devolving control. Management Science, 56(10): 1849–1872.

Boudreau, K. (2019) ""Crowds" of Amateurs & Professional Entrepreneurs in Marketplaces". NBER Working Paper No. 24512.

Bresnahan, T.F. and A. Gambardella, (1998), The division of inventive labor and the extent of the market, in Helpman, Ed., General Purpose Technologies and Economic Growth, Cambridge: pp. 253-281.

Castañer, X. (2017), Redefining Creativity And Innovation In Organisations: Suggestions For Redirecting Research, The Role of Creativity in the Management of Innovation. May: 49-71.

Cennamo, C., Ozalp, H., Kretschmer, T., (2018). Platform architecture and quality trade-offs of multihoming complements. Information Systems Research. 29 (2), 461–478.

Cennamo C, Santaló J. (2019). Generativity tension and value creation in platform ecosystems. Organization Science, 30(3), 617.641.

Clark, K. B. (1985). The interaction ofdesign hierarchies and market concepts in technological evolution. Research Policy, 14: 235–251.

Cohen WM, Levinthal DA. (1990). Absorptive capacity: a new perspective on learning and innovation. Administrative Science Quartely, 35(1): 128–152.

Corts KS, Lederman M. (2009). Software exclusivity and the scope of indirect network effects in the U.S. home video game market. International Journal of Industrial Organization, 27: 121–136.

De Vaan, M., B. Vedres, and D. Stark (2015) "Game changer: The topology of creativity." American Journal of Sociology, 120: 1144–1194.

Eaton, B.D., Elaluf-Calderwood, S. , Sørensen, C. , Yoo, Y. , (2015). Distributed tuning of boundary resources: the Case of Apple's iOS service system. MIS Quarterly, 39 (1), 217– 243.

Ethiraj, Sendil and Daniel Levinthal (2004) "Modularity and Innovation in Complex Systems," Management Science, 50(2):159.174.

Ethiraj, Sendil K., Daniel Levinthal and Rishi Roy (2008) "The Dual Role of Modularity: Innovation and Imitation," Management Science, 54(5): 939-955.

Evans, D. S., Hagiu, A., & Schmalensee, R. (2006). Invisible engines: how software platforms drive innovation and transform industries. Cambridge: MIT Press.

Flath, C.M., Friesike, S., Wirth, M., Thiesse, F., (2017). Copy, transform, combine: exploring the remix as a form of innovation. Journal of Information Technology, 32 (4), 306–325.

Fleming, L., Sorenson, O., (2001). Technology as a complex adaptive system: evidence from patent data. Research Policy, 30, 1019–1039.

Fleming L, Mingo S, Chen D (2007) Collaborative brokerage, generative creativity, and creative success. Adminstrative Science Quarterly, 52(3): 443–475.

Gambardella A, Giarratana M (2013) General technological capabil-ities, product market fragmentation, and markets for technology. Research Policy, 42(2):315–325.

Garud R, Kumaraswamy A. (1993). Changing competitive dynamics in network industries: an exploration of Sun Microsystems' open systems strategy. Strategic Management Journal, 14(5): 351–369.

Garud R, Kumaraswamy A. (1995). Technological and organizational designs to achieve economies of substitution. Strategic Management Journal, 16: 93–110.

Garud, R., Tuertscher, P. and Van de Ven, A.H. (2013). Perspectives on Innovation Processes, The Academy of Management Annals, 7(1): 775–819.

Gawer, A., 2014. Bridging different perspectives on technological platforms: toward an integrative framework. Research Policy, 43, 1239–1249.

Ghazawneh A, Henfridsson O (2013) Balancing platform control and external contribution in third-party development: The boundary resources model. Information Systems Journal, 23(2):173–192.

Hall BH, Jaffe A, Trajtenberg M. 2005. Market value and patent citations. RAND Journal of Economics, 36(1): 16–38.

Jacobides MG, Cennamo C, Gawer A (2018) Towards a theory of ecosystems. Strategic Management Journal, 39(8), 2255-2276.

Kang, Aaltonen, and Henfridsson, (2019). The Impact of Platform Entry Strategies on the Quality of Complements in Multihoming. ICIS 2019 Munich.

Kaplan S, Vakili K (2015) The double-edged sword of recombination in breakthrough innovation. Strategic Management Journal, 36(10): 1435–1457.

Karhu, K., Gustafsson, R., Lyytinen, K. (2018). Exploiting and Defending Open Digital Platforms with Boundary Resources: Android's Five Platform Forks. Information Systems Research, 29, 479-497.

Katila, R. and G. Ahuja (2002), Something old, something new: a longitudinal study of search behavior and new product introduction, Academy of Management Journal, 45(6), 1183–1194.

Kim, D.J., Kogut, B., (1996). Technological platforms and diversification. Organization Science, 7, 283–301.

Kogut, B., U. Zander. (1992). Knowledge of the firm, combinative capabilities, and the replication of technology. Organization Science, 3(3) 383–397.

Kyriakou, H., Nickerson, J., and Sabnis, G. (2017). Knowledge Reuse for Customization: Metamodels in an Open Design Community for 3D Printing, MIS Quarterly 41(1), pp. 315-332.

Majchrzak, A., Cooper, L.P. and Neece, O.E. (2004). Knowledge Reuse for Innovation, Management Science, 50(2): 174–188.

Mollick E (2012) People and process, suits and innovators: The role of individuals in firm performance. Strategic Management Journal, 33(9):1001–1015.

Mukherjee, S., Uzzi, B., Jones, B. and Stringer, M. (2016). A New Method for Identifying Recombinations of Existing Knowledge Associated with High-Impact Innovation, Journal of Product Innovation Management, 33(2): 224–236.

Nelson, R.R. and Winter, S.G. (1982). An Evolutionary Theory of Economic Change, Cambridge: Harvard University Press.

Parnas, David L. (1972) "On the Criteria to Be Used in Decomposing Systems into Modules," Communications of the ACM 15: 1053-58; reprinted in Hoffman and Weiss (eds.) Software Fundamentals: Collected Papers of David Parnas, Boston MA: Addison-Wesley.

Pil, F., C. Cohen. (2006). Modularity: Implications for imitation, inno-vation, and sustained advantage. Academy of Management Review, 31(4) 995–1011.

Rietveld J, Eggers JP (2018) Demand heterogeneity in platformmarkets: Implications for complementors. Organization Science, 29(2):304–322.

Rivkin, J.W., 2000. Imitation of complex strategies. Management Science, 46, 824–844.

Rosenkopf L, Nerkar A. 2001. Beyond local search: boundary-spanning, exploration, and impact in the optical disk industry. Strategic Management Journal, 22(4): 287–306.

Sanchez, R. and Mahoney, J.T. (1996). Modularity, Flexibility and Knowledge Management in Product and Organizational Design, Strategic Management Journal, 17: 63-76.

Schilling MA. (2000). Toward a general modular systems theory and its application to interfirm product modularity. Academy of Management Journal, 25(2): 312–334.

Schilling, M. A., and E. Green (2011) Recombinant search and breakthrough idea generation: An analysis of high impact papers in the social sciences. Research Policy, 40: 1321-1331.

Schumpeter JA. (1942). Capitalism, Socialism, and Democracy. Harper & Row: New York.

Singh J, Fleming L (2010) Lone inventors as sources of technological breakthroughs: Myth or reality? Management Science, 56(1):41–56.

Somaya, D., (2012). Patent strategy and management: an integrative review and research agenda. Journal of Management Studies, 38, 1084–1114.

Takeishi, A., T. Fujimoto. (2001). Modularisation in the auto industry: Interlinked multiple hierarchies of product, production and supplier systems. International Journal of Automotive Technology Management, 1 3–25.

Tambe, P. , Horton, J., (2019). The Death of a Technical Skill, Working Paper.

Teodoridis, F. (2017). Understanding team knowledge production: The interrelated roles of technology and expertise. Management Science, 64(8), 3625-3648.

Tiwana, A., Konsynski, B. & Bush, A. (2010) Platform evolution: coevolution of platform architecture, governance, and environmental dynamics. Information Systems Research, 21, 685–687.

Tiwana, A. (2018). Platform synergy: Architectural origins and competitive consequences. Information Systems Research, 29(4), 829–848.

Ulrich, K. (1995), The role of product architecture in the manufacturing firm, Research Policy, 24(3), 419–440.

Utterback, J., (1994). Mastering the Dynamics of Innovation. Harvard Business School Press, Cambridge.

Uzzi, B., S. Mukherjee, M. Stringer, and B. Jones (2013). Atypical combinations and scientific impact. Science, 342: 468–472.

von Hippel E, Katz R (2002) Shifting innovation to users via toolkits. Management Science. 48(7):821–833.

Wareham, J., Fox, P.B., Cano Giner, J.L.: Technology Ecosystem Governance. Organization Science, 25, 1195-1215 (2014)

Yoo Y, Henfridsson O, Lyytinen K (2010) The new organizing logic of digital innovation: An agenda for information systems research. Information Systems Research, 21(4):724–735.

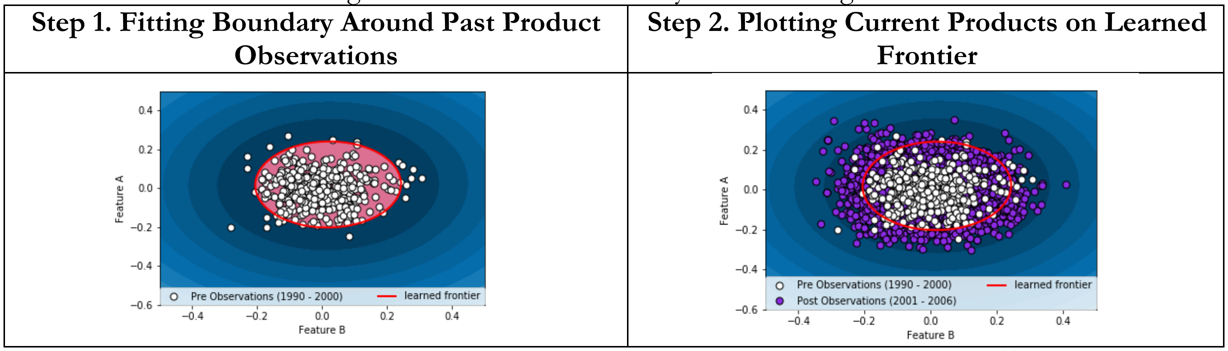
Yoo, Y. (2019). A Network Analysis on the Combinatorial Generativity of a Platform Ecosystem: The Case of WordPress. Working Paper.

Zhao, E. Y., Ishihara, M., Jennings, P. D., & Lounsbury, M. (2018). An exemplar-based model of proto-category evolution: Strategic differentiation and performance in console video game industry. Organization Science, 29, 588–611.

Zittrain, J. (2008) The Future of the Internet. Yale University Press, New Haven and London.

TABLES AND APPENDICES

Figure 1. Illustration of Novelty Detection Algorithm

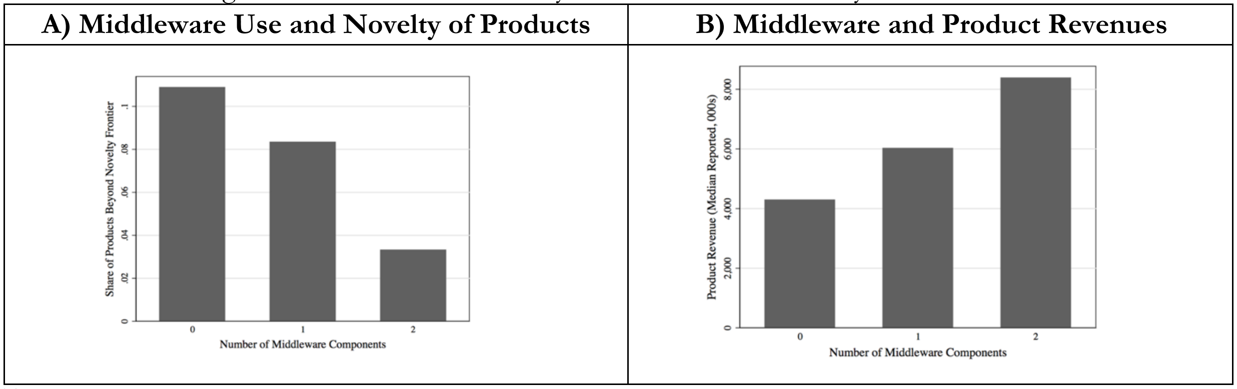


Step 1. Fitting Boundary Around Past Product Observations

Step 2. Plotting Current Products on Learned Frontier

The novelty detection algorithm used in this paper is based around a one-class Support Vector Machine Classifier. This algorithm fits a contour (boundary) based on observations at a time before the current period (ie. pre-observations). Choosing this boundary has some discretion and we can allow for a certain share of observations to be outside of the boundary (i.e. white dots outside of the learned frontier in Figure 1A). We then overlay (or predict in mathematical terms) whether the observations that come after are within or outside of the learned frontier. Those outside the frontier are considered to be novel, while those inside are considered to not be novel. The novelty score indicates the distance of a particular observation from the boundary of the learned frontier. This means that the relative measure of the novelty score between novel and less novel products will be consistent no matter the exact position of the frontier.

Figure 2. Breakdown of Novelty and Product Revenues by Middleware Use

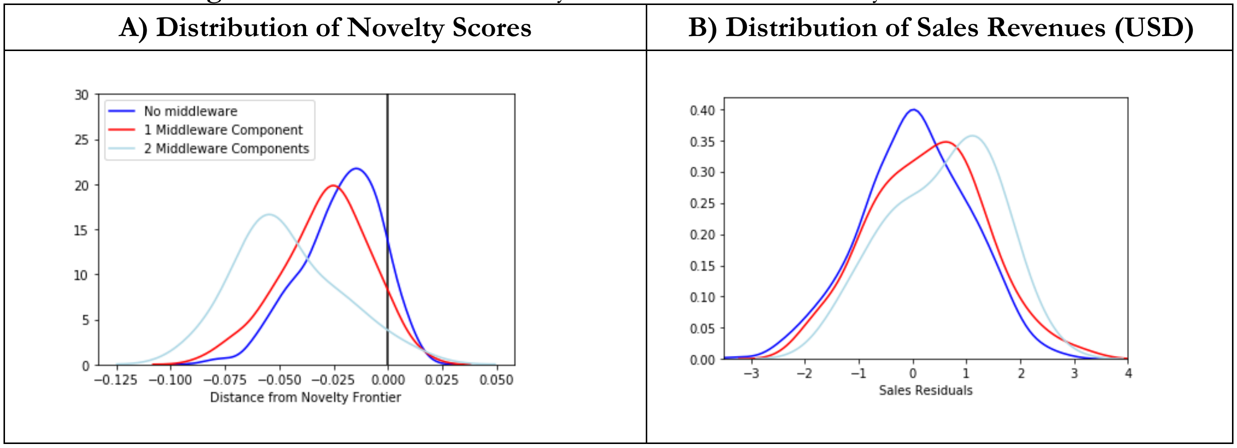


A) Middleware Use and Novelty of Products

B) Middleware and Product Revenues

The Novelty Frontier represents the boundary fitted by the one-class classification algorithm (illustrated in Figure 1). A product beyond the novelty frontier is considered to be novel. Titles beyond the novelty frontier plotted with respect to the share of middleware components used (Fig A). Also, we plot the median revenue within each group (Fig B). Median reported as a result of high variance within these groups.

Figure 3. Distribution of Novelty Score and Product Sales by Middleware Use



A) Distribution of Novelty Scores

B) Distribution of Sales Revenues (USD)

These figures provide an overview of the distribution of product novelty scores and product revenues, over the use of middleware. The distributions are shown using histograms overlayed with kernel density regressions (epanechnikov kernel). The figure on the left provides a distribution of novelty scores, with the novelty threshold represented by the vertical blue line. While this threshold changes based on the parameters of the novelty detection algorithm, the distributions relative to this threshold remain relatively constant. The blue distribution shifted to the right of the red distribution suggests that those without middleware are on average more novel. The figure on the right provides a similar breakdown based on residuals of product demand (total sales), suggesting that those that use middleware components generate lower sales on average.

Table 1. Results of OLS Regressions for Product Novelty
Unit of Observation: *Individual Title-platform*
Outcome Variable: *Product Novelty Score (i.e. Distance from Novelty Frontier)*

	(1)	(2)	(3)	(4)	(5)
N. of Middleware Components	-0.003***	-0.003***	-0.003***		
	(0.001)	(0.001)	(0.001)		
N. of Middleware Components = 1				-0.000	
				(0.001)	
N. of Middleware Components = 2				-0.014***	
				(0.003)	
Middleware Components: Game Engine					-0.003
					(0.004)
Middleware Components: Graphics Engine					-0.003
					(0.002)
Middleware Components: Physics Engine					-0.003
					(0.002)
Middleware Components: 3D Engine					-0.005**
					(0.002)
In House Middleware			0.001	0.001	0.001
			(0.001)	(0.001)	(0.001)
Project Size			-0.002**	-0.001**	-0.002**
			(0.001)	(0.001)	(0.001)
Licensed Title			0.001	0.001	0.001
			(0.001)	(0.001)	(0.001)
Middleware Experience			-0.002	-0.002	-0.002
			(0.001)	(0.001)	(0.001)
Product Experience			0.000	0.000	0.000
			(0.001)	(0.001)	(0.001)
Time FE	Yes	Yes	Yes	Yes	Yes
Publisher FE		Yes	Yes	Yes	Yes
Category FE		Yes	Yes	Yes	Yes
Intercept	-0.002	0.001	0.007	0.007	0.007
	(0.002)	(0.004)	(0.006)	(0.006)	(0.006)
F	33.03	6.82	6.68	6.84	6.55
	0.00	0.00	0.00	0.00	0.00
R^2	0.42	0.48	0.48	0.49	0.48

Robust Standard errors in parentheses. (p<.1, ** p<.05, ***p<.01) N = 1112*

Table 2. Results of OLS Regressions for Product Value (Total Sales)
Unit of Observation: *Individual Title-platform*
Outcome Variable: *Total Product Sales (Log - Transformed)*

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
N. of Middleware Components	0.356***	0.335***	0.288***	0.260***			0.248***
	(0.089)	(0.081)	(0.079)	(0.077)			(0.077)
N. of Middleware Components = 1					0.273***		
					(0.094)		
N. of Middleware Components = 2					0.482**		
					(0.221)		
Middleware Components: Game Engine						-0.335	
						(0.283)	
Middleware Components: Graphics Engine						0.158	
						(0.134)	
Middleware Components: Physics Engine						0.406***	
						(0.126)	
Middleware Components: 3D Engine						-0.335	
						(0.175)	
Distance from Novelty Frontier							-3.675
							(2.544)
In House Middleware			0.590***	0.506***	0.505***	0.524***	0.525***
			(0.076)	(0.076)	(0.077)	(0.077)	(0.077)
Project Size			0.417***	0.418***	0.408***	0.420***	0.413***
			(0.053)	(0.054)	(0.054)	(0.054)	(0.053)
Licensed Title			-0.040	-0.038	-0.037	-0.034	-0.036
			(0.079)	(0.077)	(0.077)	(0.077)	(0.077)
Product Experience			0.084*	-0.004	-0.004	-0.010	0.026
			(0.049)	(0.104)	(0.104)	(0.105)	(0.104)
Middleware Experience				0.099	0.098	0.100	0.094
				(0.099)	(0.099)	(0.099)	(0.099)
Time FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Publisher FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Category FE		Yes	Yes	Yes	Yes	Yes	Yes
Intercept	15.167***	14.088***	13.333***	12.249***	12.247***	12.241***	12.156***
	(0.154)	(0.414)	(0.427)	(0.489)	(0.490)	(0.489)	(0.490)
F	7.09	7.28	8.08	8.99	8.89	8.82	8.93
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
R^2	0.14	0.41	0.45	0.48	0.49	0.50	0.51

Robust Standard errors in parentheses. (p<.1, ** p<.05, ***p<.01) N = 1112*

Table 3. Results of Regressions for Use of Middleware Components [Robustness Check #1]
Unit of Observation: *Individual Title-platform*
Outcome Variable: *Number of Middleware Components Used*

	(1)	(2)	(3)	(4)	(5)	(6)
	Entrant Indicator		Experience with Past Releases		Only Developers with Previous Releases	Only Entrants
New Entrant	0.310*	0.331*				
	(0.142)	(0.148)				
Project Size	0.350***	0.236*	0.391***	0.266*	0.423**	0.027
	(0.104)	(0.109)	(0.105)	(0.109)	(0.155)	(0.154)
Number of Past Products			-0.247***	-0.310***	-0.438**	
			(0.070)	(0.087)	(0.135)	
Year, Publisher, Category FE & Other Controls		Yes		Yes	Yes	Yes
N	1112	1112	1112	1112	717	395
Chi-Squared	108.006	174.258	115.477	182.351	118.282	87.170
	0.000	0.000	0.000	0.000	0.000	0.000

Robust Standard errors in parentheses (p<0.05, ** p<0.01, *** p<0.001)*

Table 4. Results of Instrumental Variable Regressions [Robustness Check #2]
Unit of Observation: *Individual Title-platform*

	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
N. of Middleware Components	-0.003***	-0.006*	-0.006*	0.248***	0.581*	0.563*
	(0.000)	(0.003)	(0.003)	(0.077)	(0.293)	(0.304)
First Stage Results						
Number of Past H1B Cases		0.041*			0.041*	
		(0.016)			(0.016)	
Post H1B Policy *X Number of Past H1B Cases*		-0.051**			-0.051**	
		(0.019)			(0.019)	
Year, Publisher & Category FE	Yes	Yes	Yes	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes	Yes	Yes	Yes
Second Stage F	8.68	8.79	8.61	5.95	8.66	2.31
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
First Stage F for Excluded		4.05	2.45		3.82	2.34
(Sand.- Windm, Multivariate F Stat. Reported)		(0.01)	(0.08)		(0.02)	(0.09)
Hansens J Statistic		0.035			1.049	
		(0.85)			(0.30)	
Kleinberg Papp LM Statistic (Chi 2)		8.80			8.32	
		(0.01)			(0.01)	
R^2	0.47	0.48	0.45	0.38	0.38	0.38

Robust Standard errors in parentheses. (p<.1, ** p<.05, ***p<.01) N = 1112. Stars indicated at 0.1level to convey significance levels of observations around threshold.*

The instrumental variable regressions here are used to evaluate whether once controlling for selection the results change. We find that after introducing the instruments, our results do not change, but only weaken in significance, indicating that this our results are not really driven by the decision of which companies will use middleware, but rather driven by the features of the middleware technology itself. Our first stage outcome variable is a count ranging from zero to three, while the predicted values introduced into the second stage of the 2SLS are normally distributed continuous variables. This changes considerably the magnitude of the coefficients and they should therefore be interpreted with caution. However, they key test of whether the effect remains or is changed following the introduction of these instruments remains valid and provides support for our theoretical and empirical results. First stage F statistics for multiple instruments is reported. Kleinberg Papp statistics suggest that the instruments may be weak, and that this may bias the second stage estimates, therefore we include the *condivreg* estimates for instrumental variable regressions in Stata, which is based on the conditional likelihood approach developed by Moreira (2003) as intended for weaker instruments.

Table 5. Results of OLS Regressions for Product Novelty & Demand with Middleware Specialization [Robustness Check #3]

Unit of Observation: *Individual Title-Platform*

	(1)	(2)
Outcome Variable:	Novelty Score [Table 1]	Product Sales [Table 2]
N. of Middleware Components	0.002	0.023
	(0.002)	(0.144)
In House Middleware	0.001	0.501***
	(0.001)	(0.076)
Project Size	-0.001**	0.668***
	(0.001)	(0.084)
Licensed Title	0.001	-0.037
	(0.001)	(0.077)
Middleware Experience	-0.000	0.025
	(0.001)	(0.105)
Product Experience	-0.000	0.033
	(0.001)	(0.014)
Middleware Experience	-0.003***	1.139*
x N. of Middleware Components	(0.001)	(0.072)
Time FE	Yes	Yes
Publisher FE	Yes	Yes
Category FE	Yes	Yes
Intercept	0.006	12.213***
	(0.006)	(0.489)
F	6.76	8.96
	0.00	0.00
R^2	0.49	0.52

Robust Standard errors in parentheses (p<0.05, ** p<0.01, *** p<0.001).*

N = 1112