Check for updates

JIT

Research Paper

Journal of Information Technology
2020, Vol. 35(3) 251–269
© Association for Information Technology Trust 2020

Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0268396220936705
Journals.sagepub.com/jinf

The ownership of digital infrastructure: Exploring the deployment of software libraries in a digital innovation cluster

Lior Fink[1], Jianhua Shao[2,3], Yossi Lichtenstein[4] and Stefan Haefliger[3]

Abstract

Boundary resources have been shown to enable the arm's-length relationships between platform owners and third-party developers that underlie digital innovation in platform ecosystems. While boundary resources that are owned by open-source communities and small-scale software vendors are also critical components in the digital infrastructure, their role in digital innovation has yet to be systematically explored. In particular, software libraries are popular boundary resources that provide functionality without the need for continued interaction with their owners. They are used extensively by commercial vendors to enable customization of their software products, by communities to disseminate open-source software, and by big-tech platform owners to provide functionality that does not involve control. This article reports on the deployment of such software libraries in the web and mobile (Android) contexts by 107 start-up companies in London. Our findings show that libraries owned by big-tech companies, product vendors, and communities coexist; that the deployment of big-tech libraries is unaffected by the scale of the deploying start-up; and that context evolution paths are consequential for library deployment. These findings portray a balanced picture of digital infrastructure as neither the community-based utopia of early open-source research nor the dystopia of the recent digital dominance literature.

Keywords

Digital infrastructure, ownership, software libraries, deployment, boundary resources, software development context, entrepreneurial companies

Introduction

The importance of infrastructure for digital innovation has been repeatedly recognized (Constantinides et al., 2018; De Reuver et al., 2018; Henfridsson and Bygstad, 2013; Tilson et al., 2010). In particular, software development kits (SDKs), application approval processes, and application programming interfaces (APIs) have been identified as boundary resources that cultivate the platform business model by balancing generativity and control (Eaton et al., 2015). Boundary resources can be viewed as artefacts 'plastic enough to cut across multiple social worlds by providing enough structure to support several parties and their employed activities within separate social worlds' (Ghazawneh and Henfridsson, 2010: 4). They are the tools that serve as interfaces for arm's-length relationships between platform owners and application developers (Ghazawneh and Henfridsson, 2013). Drawing on this literature, the concept of boundary resources has been foundational to explaining the processes through which software artefacts become digital infrastructure (Eaton et al., 2015).

Despite these important contributions, there is little understanding of the role of boundary resources that are owned by software product vendors and public open-source

[1]Ben-Gurion University of the Negev, Israel
[2]Alibaba Cloud
[3]Cass Business School, City, University of London, UK
[4]Queen Mary University of London, UK

Corresponding author:
Lior Fink, Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, 1 Ben-Gurion Street, Beer-Sheva 84105, Israel.
Email: finkl@bgu.ac.il

communities rather than by platform owners. As boundary resources bind infrastructure owners of various types with deployers, the issue of ownership is key to understanding digital infrastructure. Indeed, new digital industries, emerging technological monopolies (Khan, 2018; Warren, 2019), and possibly a new form of capitalism (Zuboff, 2015, 2019) are based on a mesh of private and public resources (Yoo et al., 2010a). In this context, the boundaries between private and public are often blurred, as monopolies stack private layers on the layered modular architecture of public standards (Crémer et al., 2019; Pon et al., 2014) while releasing privately developed technologies as public and free open-source software (OSS) (Rock, 2019). Despite the importance of infrastructure for digital innovation and the complexity of its ownership, little research has been conducted on the implications of the ownership of boundary resources.

Focusing on the deployment of digital infrastructure, the purpose of this article is to study the deployment of software libraries (Flath et al., 2017; Haefliger et al., 2008) contingent on their ownership. Namely, the study aims at addressing the following research question: *How does the ownership of software libraries affect their deployment?* A software library is a standard way of packaging code that becomes a boundary resource when it is offered outside of an organization. Software libraries are consistent with the bidirectional causality of boundary resources (Ghazawneh and Henfridsson, 2013), as they are designed to advance the goals of both owners and users. However, the trade-offs underlying these resources are different for communities (innovation vs coherence), software vendors (complexity vs flexibility), and large platform owners (generativity vs control). These distinct trade-offs suggest that the deployment of software libraries is affected by who owns them: communities, proprietary software vendors, or big-tech providers.

To answer this research question, we observe the deployment of web and mobile (Android) software libraries by 107 start-ups operating in London, one of the largest European digital innovation clusters. Our empirical analysis confirms that ownership is consequential for software library deployment, which is more significantly affected by the software development context (web or mobile) than by the scale of the deploying company (its stage of growth). This study contributes to the literature on digital infrastructure in several ways. Conceptually, this work extends previous research on digital infrastructure (Constantinides et al., 2018; De Reuver et al., 2018; Tilson et al., 2010) by considering software libraries as boundary resources (Ghazawneh and Henfridsson, 2010, 2013), the deployment of which is contingent on whether they are owned by communities, proprietary vendors, or big-tech providers. In so doing, we extend the classical distinction between public and private resources by conceptualizing resources owned by dominant digital providers as a third ownership type. Empirically, this work sheds light on the revealed preferences of infrastructure users as reflected in the deployment of software libraries by start-ups operating in a major digital innovation cluster. Altogether, our findings portray a novel and balanced picture of the current digital infrastructure and the nuanced boundaries between public and private resources.

Theoretical foundations

Ownership of software

The ownership of information and digital artefacts is theorized in the OSS literature as a dichotomy between proprietary and commons. The concept of *commons* refers to a resource that is shared by a group of people, originally applied to physical resources, that are both small scale (grazing land) and large scale (ocean fisheries) (Ostrom, 2015). The connection between commons and information has been advocated since the mid-1990s because information is both an input and an output of its own production and because of its zero marginal cost (Benkler, 1998; Hess and Ostrom, 2007; Lessig, 1999). Consequently, a positive price on information, once it has been produced, leads to underutilization and underproduction of new information (Benkler, 2013).

Indeed, this dichotomy was apparent during the early years of digital technology. The free software movement began in opposition to the separation between hardware companies and emerging software companies, which allowed software to be produced and sold separately from the machines that ran the software (Levy, 1994). Richard Stallman articulated this idea and the ideology that continues to drive OSS movements on the basis of norms, beliefs, and values specific to the community (Stallman, 1985; Stewart and Gosain, 2006). These values include the sharing of code as a public good; helping and cooperating with others; and freely using, changing, and redistributing code (Levy, 1994; Stallman, 1985). OSS licences, and before them the GNU General Public License (GPL), have been designed to make software a public good and to disseminate and share programmes as widely as possible among users and developers. In fact, the viral characteristic of free software licences, such as the GNU GPL, mandates that all software used with free software becomes a public good as well (Moody, 2001). OSS is similar to privately produced public goods in its lack of excludability by virtue of the licence and lack of rivalry as a knowledge good. However, it differs from traditional public goods by being owned by groups of individuals who guard their reputations as contributors (O'Mahony, 2003).

In complete opposition to this logic of comprehensive sharing, and in alignment with the capitalist norms of Western society, the software industry endorses a

proprietary mode of ownership similar to almost every other industry (Cusumano and Selby, 1998). Software is developed and compiled in a format only readable to machines and is sold or licenced by its owner. Extensive customer input defines the requirements of software to be developed, and the development process follows a vision statement that defines the goals for a new product that best serves customers (Brooks, 1975; Cusumano and Selby, 1997). Development teams come up with features that typical customers may appreciate, and they observe and interact with customers repeatedly to update requirements. For proprietary software, the development methodology involves a customer, and the outcome is expected to fit customer needs precisely and narrowly (Sommerville, 2000).

In addition to these two modes of ownership, we suggest that a third, hybrid mode has gained importance over the last decades. We label this mode *big-tech* because it has emerged with respect to digital platforms (Jacobides et al., 2018; Parker and Van Alstyne, 2005), such as those owned by Microsoft, Google, and other dominant technology providers. Microsoft serves as a good example because its products have been fundamentally proprietary while some infrastructural aspects of the Windows platform have been opened through disclosures of APIs and communication protocols (U.S. v. Microsoft Corporation, 2002). Similarly, the Google mobile platform includes the Google Play app store as a proprietary gatekeeper, while the Android operating system is open (Pon et al., 2014; Siegele, 2018). These combinations guarantee some openness for the complementors who make up the bulk of the functionality of the platforms. This third mode of ownership represents a combination of proprietary infrastructure, which allows access only through the owner, and commons infrastructure, which entails no such asymmetric power at a single point but rather a set of symmetric rules concerning access, use, extraction, and management (Benkler, 2013, 2016). Our focus in this work is on digital platforms that attract large numbers of complementors and are owned by big-tech companies (Cusumano, 2012; Gawer and Cusumano, 2002). These companies cater to large numbers of independent software vendors and nourish their large horizontal communities (Fitzgerald, 2006), achieving their goals with respect to their direct customers and wider communities of complementors.

The release of deep learning software packages by Google (TensorFlow) and Facebook (PyTorch) is a point in case; big-tech companies navigate their proprietary interests to serve their customers directly and, at the same time, expose their technology to expand the user base of critical components that can attract complementors or talent (Alexy et al., 2013; Rock, 2019). The role of big-tech companies as platforms that mediate between customers and community members alters their behaviour as owners of software.

In summary, we follow the OSS literature to contrast proprietary (company-owned) software with community (commons-governed) software. To address the growing power of dominant companies (Eaton et al., 2015), we further distinguish between proprietary software that is owned by big-tech companies and proprietary software that is owned by less dominant vendors. Given our focus on software libraries, we distinguish between community, proprietary, and big-tech software libraries according to the ownership mode characterizing their providers.

Software libraries as boundary resources

Technological building blocks become infrastructure when taking on the role of a boundary resource between users and owners. Specifically, we focus on software libraries by extending the subset of resources that have been studied regarding Apple's infrastructure, namely SDKs, APIs, and developer agreements (Ghazawneh and Henfridsson, 2010, 2013). A software library is a standard way of packaging code either within an organization or for external utilization. The latter is used extensively by platform owners, software vendors, and OSS communities, implying that libraries assume the role of boundary resources between users and owners of software. A software library is deployed to implement a specific functionality without the need for explicit recoding, suggesting that it is similar to an API in the sense that it facilitates quick integration of functionality (Kim and Stohr, 1998). However, in contrast to an API, a software library includes code that provides functionality and, therefore, does not require continued interaction with the library's owner to attain that functionality. Software libraries are critical to innovation because they enable low-cost experimentation, learning, and creation of new functionality through the reuse of existing resources (Flath et al., 2017; Haefliger et al., 2008) and human problem-solving (Barns and Bollinger, 1991).

The role of a software library is often contingent on its ownership. OSS communities provide libraries that cover the full functionalities of their software, aiming for full and open distribution. By contrast, software product vendors typically provide libraries that include the limited functionality necessary to customize the product to a customer environment, such as libraries that connect to different information technology (IT) configurations at the customer site and allow the product to process the customer's data. Libraries provided by platform owners enable functionality that requires no access to external resources and provides, for example, access to graphics or data. Such functionality is considerably quicker and simpler than that enabled by APIs that access external resources, such as the Global Positioning System or in-app purchases (Ghazawneh and Henfridsson, 2013).

From a boundary resource perspective, software libraries serve as repositories, coincident, and ideal objects, similar to APIs (Ghazawneh and Henfridsson, 2010). Libraries are repositories because they are a reference for data

structures, object classes, protocols, and so on. Similar to APIs, libraries help developers solve problems by providing shared definitions and values. Libraries are also coincident boundary resources, as they offer users a common referent for using their different functionalities independently. Finally, libraries are ideal boundary objects because, similar to APIs, they support many settings without detailing specific usage.

Software libraries also fit the bidirectional causality of boundary resources (Ghazawneh and Henfridsson, 2013). They are designed to advance the goals of both owners and users, and their design and use incorporate feedback mechanisms and mutual shaping. Recent literature has focused on platform owners and complementors, suggesting that the emergence of a new API can be the result of the identification of new external contribution opportunities by a platform owner as well as the outcome of a proactive request from a third-party developer (Ghazawneh and Henfridsson, 2013). The same applies to software libraries. Whereas a software product vendor may provide a library to its customers to enable customization or other product-related functionality, customers may request a library to implement new and dynamically customized functionality. Similarly, while an OSS community usually provides a large set of libraries to enable the functionality it has implemented, users often request the community to provide additional functionality and libraries.

Conceptually, this constant process of production, reproduction, and transformation of boundary resources (Eaton et al., 2015) necessitates distinguishing between ownership types because infrastructural relations are formed from different vantage points. Big-tech companies tread a fine line between growing their third-party developers through generativity and maintaining control over the platform (Tiwana et al., 2010). Their boundary resources, including software libraries, are designed to cultivate the platform business model by balancing generativity and control. Proprietary vendors wish to fulfil customer requirements for their products and, to operate sustainably, they trade-off product complexity for product flexibility and reusability. Communities balance innovation and coherence as they attempt to avoid the fragmentation of their user and developer bases as a consequence of innovating too radically (Kogut and Metiu, 2001). Namely, for each type of ownership, the owner and deployer benefit when the trade-off can be maintained, that is, when the boundary resource optimally balances generativity and control (for big-tech ownership), complexity and flexibility (for proprietary ownership), and innovation and coherence (for community ownership).

Deployment of software libraries

Following the conceptualization of software libraries as boundary resources, which are differently designed for different ownership modes, we now focus on the users of software libraries and on their sensemaking and organized practice (Henfridsson and Bygstad, 2013; Vaast and Walsham, 2009). Star and Ruhleder (1996) explained that the common metaphor of infrastructure as a ready-to-hand, completely transparent artefact is misleading. They argued that infrastructure is a fundamentally *relational* concept, as it becomes infrastructure in relation to organized practices; for example, the cook considers the water system as infrastructure integral to making dinner. Infrastructure is, therefore, embedded in dense sociotechnical imbrications that shape and are shaped by a community of practice (Karasti and Blomberg, 2018; Star and Ruhleder, 1996). Namely, the infrastructural relations between owners and users of software libraries are constantly produced, reproduced, and transformed (Eaton et al., 2015), connect and scale up (Ribes and Lee, 2010), become embedded in practices (Star and Ruhleder, 1996), or enacted (Karasti and Blomberg, 2018).

For example, a company developing artificial intelligence (AI) applications (a practice) may use the Linux operating system and its software libraries to interact with different devices, including a laptop video camera. The new application may then access pictures in a database via a library provided by the NoSQL Oracle product (suitable for pictures) and access a face recognition software library (in addition to an API) provided by Microsoft's cognitive services platform. In making such deployment decisions, the company considers the functionality of each of these infrastructural systems; the software libraries, APIs, and SDKs they provide; their stability and flexibility; and, importantly, the long-term relations with their owners (Linux, Oracle, and Microsoft) and the control those owners may impose over the commercial deployment of the new AI application. There are quite a few alternatives that the company may consider, including Windows and Android as operating systems and their video libraries, as well as numerous OSS and commercial database management systems. Alternative implementation paths may also be considered, including operating system agnostic video libraries and deep neural network OSS libraries, such as Google's TensorFlow or Facebook's PyTorch, instead of Microsoft's new platform. The company's past experience with these providers and their infrastructure, its experience with commercial licencing versus OSS communities, and its own scale and resources are all relevant considerations for choosing among the different alternatives.

The considerations discussed above, including the trade-off that the owner of the boundary resources aims at maintaining, the key characteristics (identity and plasticity) of the boundary resource, and the key considerations (heterogeneity and autonomy) of the deployer, are summarized in Table 1. The plasticity of boundary resources, or their malleability, reflects the degree to which the digital artefacts are able to perform additional and extensible functions

Table 1. Considerations related to software libraries as boundary resources.

Consideration	Ownership		
	Proprietary	Big-tech	Community
Trade-off maintained by owner	Product complexity vs product flexibility and reusability	Generativity for third-party developers vs control by the platform owner	Innovation vs stability and coherence
Identity of boundary resource	Part of a software product	Part of a digital platform, often in conjunction with the owner's applications	Part of an OSS project
Plasticity of boundary resource	Binary code; can be used in any customer context but cannot be changed	A combination of binary and open-source code; maintained by the owner, but minor changes by the deployer are allowed	Open source; can be used in any context and can be changed as needed; well-maintained by large, active communities
Deployer heterogeneity	Only direct customers of the underlying product	Third-party developers as well as direct customers	The wide public, but usually developers of downstream systems and products
Deployer autonomy	Any use, constrained by the terms and conditions of the underlying product	As governed by additional APIs and agreements	Any use

OSS: open-source software; API: application programming interface.

above and beyond their original purpose (Yoo et al., 2010b). While the functions of binary-deployed (or closed) software are extended by how it is deployed and used, OSS enables the reshaping of the code itself to significantly increase the options for adding and extending the functionality of the software.

Hypotheses

Drawing on the theoretical foundations laid out above, we develop three hypotheses about the relationships between ownership and deployment of software libraries. The complexity of the software landscape serves as a basis for forming an initial expectation that companies deploy software libraries that represent all ownership types: community, proprietary, and big-tech. The relational perspective of infrastructure is then employed to formulate two hypotheses: that the deployment of big-tech libraries is independent of company scale (i.e. companies of different scales similarly deploy big-tech libraries) and that companies of different scales demonstrate different preferences for community and proprietary libraries. The third hypothesis considers two important contexts of library deployment – web and mobile – which are used as the empirical setting of this study. Specifically, we hypothesize that companies that operate in both contexts exhibit context-specific preferences for community and proprietary libraries.

A basis for the hypotheses is the expectation that companies that introduce digital artefacts into the market deploy infrastructure that originates from various sources representing all types of infrastructure ownership. This expectation is grounded in the diversity of the goals and needs of deploying companies, the breadth of infrastructural components, and the rationale that this complexity is accommodated by the full gamut of ownership types. The software landscape has become a complex mesh of different communities and companies, and it is now practically impossible to deploy infrastructure from only a single source or ownership type. When a specific functionality is available from multiple sources, dominant proprietary and big-tech providers typically rank high on quality and stability, but their binary code libraries are less flexible than the open-source code provided by communities. Beyond boundary resource plasticity, community libraries are also superior to proprietary and big-tech libraries in terms of cost and stability, as community libraries are free and usually very stable, while large and small providers are bound to change technological paths and to charge for their infrastructure, even if it was initially free. Proprietary providers, however, are able to offer libraries that better cater to the needs of the companies that use their software products. Given the heterogeneity of deploying companies and their development teams, there are multiple optimal points for the trade-off between these considerations, and therefore, we expect the three ownership types to cohabitate. In other words, for a specific functionality and a given set of available libraries, some companies prefer the plasticity and autonomy of community libraries, some prefer the quality and stability of libraries owned by big-tech providers, and others prefer the customization and fit offered by libraries owned by smaller providers.

A similar reasoning was expressed by Lerner and Schankerman (2010), who surveyed more than 2000 business and government users of software in 15 countries and found that OSS and proprietary software were mixed or comingled. Their economic analysis offered three main reasons for this finding. First, OSS and proprietary software pose different cost trade-offs to users; OSS is inexpensive

to deploy but requires greater costs for switching, interoperability, learning, and support. Second, users are highly heterogeneous in terms of their needs and the weight they attach to different types of costs. Third, larger, higher technology users are more likely to extensively mix OSS and proprietary software. In summary, given the heterogeneity of the deploying companies, the complexity of infrastructure, and the variety of optimal points in infrastructure trade-offs, we expect to find that companies deploy software libraries that represent all ownership types: community, proprietary, and big-tech.

The first hypothesis addresses the role of big-tech in infrastructure deployment. The recent critique of dominant digital companies (Benkler et al., 2018; Moore and Tambini, 2018) has focused on economic, anti-trust, political, and media aspects. However, as dominant digital companies maintain and protect their positions by embedding customers within their networks (Barwise and Watkins, 2018), they also embed downstream providers within their infrastructural services. This aspect has been studied for digital platforms. Eaton et al. (2015), for example, warned that any theoretical attempt to deal with such innovation must deal with the tension between digital generative and democratizing forces and the 'monopolistic and controlling force of digital infrastructure' (p. 218). Ghazawneh and Henfridsson (2013) showed that boundary resources shape the relationship between a platform owner and application developers. We argue that for any boundary resource, whether platform or not, dominant players translate and inscribe their dominance into technology, specifically infrastructural components. This dominance is first manifested in the single software development context, as analysed by Eaton et al. (2015) and by Ghazawneh and Henfridsson (2013). Then, dominance in one context may be transferred to other contexts. For example, Google's search engine dominance of the web has enabled it to become one of the two dominant digital players in the mobile context, and it is now an important player in machine learning (Rock, 2019). However, dominance in one software development context is not always transferable to other contexts, as the history of the dominant companies of the mainframe and mini-computer eras shows. Full development of this line of reasoning requires consideration of the platform, ecosystem, and community of practice aspects of infrastructure, which are beyond the theoretical scope of this work.

In our research setting, digital dominance is expected to be manifested as stable demand across different company scales. The concept of *scale* is used to distinguish between different stages of growth in the evolution of entrepreneurial companies. When a small start-up considers possible libraries for a specific task, a big-tech library represents a safe, low-risk, and high-quality choice, which is frequently considered the de facto standard. The downside of lower plasticity is acceptable at this stage in the company's lifecycle. Importantly, big-tech providers enable small companies to deploy libraries for free through relatively long trial periods and free run-time operation for small- to medium-size volumes. For larger companies, a big-tech library represents a good compromise. On one hand, resource plasticity is an issue because the boundary resource is provided as a binary code that cannot be modified, and deployer autonomy is an issue because use of the boundary resource is restricted. Furthermore, payment is usually required given the large volume of run-time operations typical of such companies. On the other hand, the stability, quality, and security of the library compensate for the lower plasticity and autonomy it imposes. In summary, we argue that software libraries owned by dominant technology companies are needed by both small- and large-scale companies. Therefore, we expect to see relatively little variance in the deployment of big-tech libraries across different company scales:

> *Hypothesis 1 (big-tech stability).* Companies of different scales similarly deploy big-tech libraries.

The second hypothesis considers libraries that are not provided by dominant digital infrastructure suppliers. It addresses differences in the deployment of community and proprietary libraries across companies of different scales by drawing on the concept of experimentation from the OSS literature. We reason that smaller companies are highly likely to facilitate innovation by experimenting during the development of new products and services. An important reason to deploy community libraries is the lower control of the community owner over the infrastructure, because registration, contracts, and APIs are not required and libraries are not otherwise controlled. In addition, the ability of users to inspect the code and, if necessary, to copy and change it (Lee and Cole, 2003; Shaikh and Vaast, 2016; Zeitlyn, 2003) incentivizes companies that value experimentation to deploy these libraries. These companies have few resources at their disposal and, therefore, they seek to deploy software libraries to acquire functionality quickly and cheaply. Thus, experimenting with such libraries is almost cost free (Benkler, 2013, 2016). Although there is an opportunity cost, as other libraries that may be found later to be of higher quality are not used, community libraries have a good reputation, and the opportunity cost is often considered reasonably low. Ultimately, smaller companies can benefit from the higher plasticity and heterogeneity of community libraries, which afford the autonomy that is critical for their successful growth.

By contrast, larger companies have already gained considerable knowledge about product and service offerings and about the digital infrastructure that best supports their development, implying that experimentation is less crucial for these companies. Moreover, these companies possess more resources and, therefore, should be less sensitive to

the price of infrastructure. They are more sensitive to the shortcomings of community libraries, in particular the risk of lower quality and inferior ongoing maintenance as well as no guarantee for support, at least for small, less active OSS projects. Larger companies have an installed base of customers as well as existing technical architecture and technical debt (Woodard et al., 2013) that experimentation may disrupt – something that smaller companies need not worry about. Consequently, larger companies are better positioned to benefit from the customization and fit provided by proprietary libraries:

> *Hypothesis 2 (scale dependence).* Small-scale companies demonstrate a stronger preference for community libraries over proprietary libraries relative to large-scale companies.

Finally, the notion of the *software development context* is introduced to hypothesize that in addition to being scale dependent, library deployment preferences are also context dependent. The term context is chosen to focus on infrastructure itself and to separate the analysis from relevant higher-level concepts, such as communities of practice (Karasti and Blomberg, 2018; Star and Ruhleder, 1996), digital platforms (De Reuver et al., 2018; Rolland et al., 2018), and innovation ecosystems (Jacobides et al., 2018). We assume that companies develop digital artefacts either because these are the new products or services they introduce to markets, or because digital technology is used to reach customers. We also assume that companies deploy the available infrastructure, including the software libraries we study, in each context.

The first context we study is the World Wide Web, which has been established as a collection of standards governed by not-for-profit international bodies. The creation of and access to websites is open to all using communication infrastructure that is often commercial, but governed by symmetric access rules (Benkler, 2016). Many popular web tools and their related libraries are OSS, including the Apache and Nginx web servers, the jQuery JavaScript and Angular frameworks, and the PHP programming language. Since its inception, however, companies such as Google, Facebook, Amazon, and Alibaba have become dominant on the web (Moore and Tambini, 2018). When creating websites, many companies need to use software libraries provided by these dominant companies to increase their webpage ranking, link to social networks, monetize on advertisements, and so on. Therefore, the web itself is currently a mixture of community, proprietary, and big-tech resources. Given this characteristic, companies are expected to deploy software libraries of all ownership types in the software development context of the web.

The second software development context we study is Android, which also involves a mixture of resources. Big-tech ownership, however, has become a central characteristic of this context (Pon et al., 2014; Siegele, 2018). Whereas the Android operating system is open source, Google Play, the dominant app store, serves as a proprietary gatekeeper. To pre-install Google Play, device makers are required to make Google Search the default search service and to install only Google's standard version of Android on each and every one of their models (European Commission, 2016, 2018). Research has suggested that Google has transferred significant functionality to its proprietary system, Google Play Services, and to related software libraries while reducing investment in the open-source Android operating system (Pon et al., 2014).

Given these paths of evolution, we expect to see context-contingent library deployment, even within companies. The minimal functionality of the original web protocols (Berners-Lee et al., 1992; Siegele, 2018) created a need for more libraries, and the community-oriented path of the web increased the ability of community libraries to provide the required functionalities. By contrast, the mobile context was created with more significant functionalities by its big-tech originator, implying that community libraries are less needed. In addition, the popularity of mobile commerce induced many independent software vendors to provide products and services for this context. The result is that in companies that operate in both contexts, we expect community libraries to be preferred in the web context and proprietary libraries in the mobile context. This expectation is based on the alignment between the path by which a context evolves and the boundary resources that best cater to the needs of users in that context. Because plasticity, heterogeneity, and autonomy have been more (less) dominant in the evolution of the web (mobile) context, community (proprietary) libraries should be preferred in this context:

> *Hypothesis 3 (context dependence).* Companies that operate in both web and mobile contexts demonstrate a preference for proprietary libraries over community libraries in the mobile context relative to the web context.

Empirical analysis

Research setting

Empirically testing research hypotheses regarding the deployment of software libraries by entrepreneurial companies is challenging for three primary reasons. First, despite the significant advancement of the literature on digital infrastructure in recent years, its methodological basis remains relatively narrow, frequently utilizing qualitative or case-based methods. Given this methodological orientation of previous work, the use of quantitative methods to test research hypotheses requires an exploratory approach, which places more emphasis on which variables are important and how they can be measured than on considerations

of representativeness and control. Only once such variables and measures are identified should research advance to the study of richer models that take into account multiple contextual variables in cross-sectional settings. Second, software libraries, as low-level components of digital infrastructure, are typically interwoven into digital artefacts in an intricate manner. Consequently, their deployment is not easily susceptible to empirical observation, and it is extremely difficult to objectively observe their usefulness or their impact on the usefulness of the artefacts in which they are embedded. Third, the processes underlying the evolution of entrepreneurial companies vary significantly across technological contexts, innovation ecosystems, and geographic locations. This characteristic of entrepreneurial companies increases the difficulty of determining sample inclusion and exclusion criteria as well as controlling for contextual variance.

Against these empirical challenges, our approach is to focus on the deployment of software libraries by entrepreneurial companies in London, an important digital innovation cluster (Feldman and Kogler, 2010). We are able to access a subset of approximately 100 companies in this cluster that were selected on the basis of objective business criteria. The existence of this independently selected subset of entrepreneurial companies in London allows us to address the empirical challenges discussed above while minimizing contextual variance. The literature on the geography of innovation suggests that 'innovation has a decidedly geographic dimension that affects economic growth and technological change' and that the processes underlying innovation ecosystems are nuanced, subtle, pervasive, and not easily amendable to measurement (Feldman and Kogler, 2010: 383). Therefore, our approach overcomes the need to control for contextual variance at the cost of lower external validity.

Specifically, London is a leading global centre for many industries, including finance, retail, and pharmaceuticals, as well as for such creative industries as publishing, film and television, music, advertising, and fashion. The UK's Office of National Statistics estimates that in 2016 (at the time of data collection), London had approximately 300,000 digital jobs, 200 coworking spaces, 1000 start-ups being established annually, and 14 digital unicorns (start-ups valued at over US$ 1B). During that year, the sub-market of smaller companies at the London Stock Exchange raised almost £1B in 39 initial public offerings (IPOs), while London companies attracted £2.2 B of venture capital and private equity funding (Tech City UK, 2017).

Data and measures

We study all of the companies included during 2016 in the three programmes managed by Tech City, a government-funded start-up accelerator that aims at scaling up the UK's digital economy. The three programmes were (1) Upscale, which included 30 companies with over £1 M in revenue and 20% monthly growth in a key business indicator; (2) Future Fifty, which included 50 companies with revenues from £3 M to £30 M and over 30% annual revenue growth; and (3) Future Fifty Alumni, which included 27 companies that graduated from the Future Fifty programme. The exogenous classification of these 107 companies into three distinct programmes provides a valid basis for measuring the variable of scale and for testing Hypotheses 1 and 2. These companies cover a large variety of digital businesses, including computer software; financial services; electronic commerce; marketing and advertising; telecommunications; and health, wellness, and fitness. Digital infrastructure is important for all of these companies, either because their innovation is technological or because digital platforms are critical for their engagement with customers. Sample characteristics are described in Table 2. At the time of data collection (August 2016), the mean age of the 107 companies was 7.84 years, with a standard deviation of 4.46 years and a range of 0.93–24.18 years.

For these 107 companies, we collected data during the first week of August 2016 about the software libraries that each company deployed in its website and mobile apps. These data included the numbers of software libraries that were considered community, proprietary, and big-tech on both companies' websites and mobile apps. Therefore, we recorded six values for each company: one for each possible combination of context (web or mobile) and ownership (community, proprietary, or big-tech). For web libraries, we used an application detection utility called Wappalyzer.com. This utility scans a website's HTML, JavaScript, and CSS code as seen by a web browser and then analyses this code to identify software libraries by comparing elements of the code with a database of code patterns related to 820 popular libraries. This database is kept up to date by scanning millions of websites on a monthly basis (more than 200 million websites in 6 months). The analysis procedure, which is open source (github.com/AliasIO/Wappalyzer), was used to automatically scan and analyse the websites of the companies under study. The procedure was repeated several times for each company to ensure the reliable detection of libraries related to dynamic content or other dynamic configurations. Our technique is related to those used by Spinellis and Giannikas (2012) to determine the web browsers, web servers, and operating systems used by Fortune 1000 companies. Similar techniques were reported by Greenstein and Nagle (2014) in their study of web servers as well as in studies of Internet crime (e.g. Kuwatly et al., 2004). The providers with the most web library deployments in our data set are presented in Table 3.

For mobile app libraries, we automatically scrapped data from AppBrain.com. Tracking Android apps since 2010, this company regularly scans the binary code and network operation of apps and reports on the usage statistics of 353 libraries. This technique is similar to those

Table 2. Sample characteristics.

Characteristic		Scale			Total ($n = 107$)
		Upscale ($n = 30$)	Future Fifty ($n = 50$)	Future Fifty Alumni ($n = 27$)	
Industry	Digital	23	34	20	77
	Finance	4	9	2	15
	Retail	2	7	4	13
	Biotechnology	1	0	1	2
Size (number of employees)	1–10	3	0	0	3
	11–50	17	7	0	24
	51–200	10	27	9	46
	201–500	0	14	10	24
	>500	0	2	8	10

Note. The values presented in the table are frequencies. The digital industry includes hardware, software, games, IT services, and e-learning. The finance industry includes investment, lending, currency exchange, and money transfer. The retail industry includes coffee, flowers, white goods, and food delivery.

Table 3. Providers with the most web library deployments in the data set.

Provider (ownership type)	Libraries	Deployments in data set
Google (big-tech)	DoubleClick Floodlight, Tag Manager, reCAPTCHA, Maps, YouTube, Angular Material, AngularJS, Font API, DoubleClick for Publishers, Analytics, PageSpeed, AdSense	296
Oracle (big-tech)	Web Cache, Java, Java Servlet, Commerce, HTTP Server, ATG Web Commerce, AddThis, JavaServer Pages	213
Microsoft (big-tech)	SharePoint, ASP.NET, Windows Server, IIS	87
Mixpanel (proprietary)	Mixpanel	82
Chris Wanstrath and other contributors (community)	Moustache	62

used in the literature to study the app economy. In a survey of app store analyses for software engineering, Martin et al. (2017) identified 187 papers that used data from app stores or apps' actual binary codes. Although most of this research was conducted on Android apps using data collected directly from Google Play, AppBrain data has also been frequently used in the literature. The providers with the most mobile library deployments in our data set are presented in Table 4.

Libraries are classified by their ownership as follows. Libraries that are governed by communities and typically licenced as open source are classified as community. All other libraries, including those that are licenced commercially or governed by a company/individual, are classified as proprietary or big-tech. Big-tech providers are those that own digital platforms with large numbers of users and complementors. We use market capitalization as a proxy for the size of the served user base. Specifically, libraries that are governed by a company with a market capitalization above US$10 B at the time of data collection are classified as big-tech, and libraries that are governed by a company with a market capitalization below this threshold are classified as proprietary. Big-tech libraries include those provided by Google, Microsoft, Amazon, Facebook, Samsung, Oracle, IBM, SAP, Baidu, Adobe, PayPal, Yahoo, and Twitter. Libraries provided by other companies, such as Splunk, Square, and Dropbox, are considered proprietary libraries. Importantly, the three ownership classifications are mutually exclusive (i.e. a library can only have one classification).

The US$10 B threshold is used to distinguish between big-tech and proprietary companies for four reasons. First, this threshold is commonly used in the literature to define companies with high market capitalization, including in both high tech (Kuratko et al., 2020) and banking (Bouwman et al., 2018), as the Dodd–Frank Act imposes greater regulatory requirements on institutions above this threshold. Second, a threshold that is a power of 10 is preferable to mitigate possible concerns about observer-expectancy bias in choosing the threshold (i.e. to alleviate concerns that the choice of threshold may be affected by the expected results). The US$10 B threshold is chosen because the US$1 B threshold is too low (any unicorn will be classified as a big-tech company) and the US$100 B threshold is too high (relative to accepted definitions of high market capitalization in the literature). Third, from the perspective of face validity, two of the authors examined the classifications resulting from alternative thresholds and independently concluded that the US$10 B

Table 4. Providers with the most mobile library deployments in the data set.

Provider (ownership type)	Libraries	Deployments in data set
Google (big-tech)	Android Support library, Cloud Messaging, gson, Analytics, Guava, GData, Guice, Protocol Buffers, Play Games Services, Play In-app Billing	248
Square (proprietary)	okHttp, Okio, Picasso, Retrofit, Dagger, LeakCanary	144
Jake Wharton (proprietary)	NineOldAndroids, ViewPager, Butter Knife, Actionbar Sherlock, Disc LRU Cache	87
Apache (community)	Commons Codec, HttpMime API, Commons I/O, Http Auth, James Mime4j, Commons Logging, Commons Lang, Thrift	83
Fabric (proprietary)	Twitter Kit, Crashlytics, Fabric	60

Table 5. Descriptive statistics for the numbers of software libraries.

Group	Platform – ownership	Number of libraries		
		Mean	Std. dev.	Max.
Web-only companies ($n = 62$)	Web – community	2.774	1.498	7
	Web – proprietary	3.484	2.461	9
	Web – big-tech	2.597	1.741	7
Web-mobile companies ($n = 45$)	Web – community	2.467	1.272	5
	Web – proprietary	3.200	2.573	11
	Web – big-tech	2.600	1.763	8
	Mobile – community	0.778	0.951	4
	Mobile – proprietary	6.356	5.913	24
	Mobile – big-tech	3.156	2.225	9
All companies ($n = 107$)	Web – community	2.645	1.409	7
	Web – proprietary	3.364	2.501	11
	Web – big-tech	2.598	1.742	8

Note. Web-only companies deploy no mobile libraries and, therefore, descriptive statistics for mobile – community, mobile – proprietary, and mobile – big-tech libraries are not presented for web-only companies and for all companies. The minimum number of libraries in all cases is 0.

threshold most validly identifies companies commonly considered to be big-tech. In addition, the largest proprietary (non-big-tech) providers in our data set, such as Dropbox or Splunk, are focused on products, and their freemium business models (Osterwalder and Pigneur, 2010) are centred on these products rather than on digital platforms. Fourth, a sensitivity analysis demonstrates that even if a US$100 B threshold is used instead of the US$10 B threshold, the classification will change only for a single provider (Twitter will no longer be classified as big-tech).

The data collection shows that all 107 companies have a web presence and can deploy community, proprietary, or big-tech libraries in the web context. Of these companies, 62 have no mobile apps. Naturally, these 'web-only' companies cannot deploy community, proprietary, or big-tech libraries in the mobile context. The remaining 45 companies have at least one mobile app. These 'web-mobile' companies can deploy libraries across the three types of ownership and two contexts. Table 5 presents descriptive statistics for the numbers of software libraries.

Analysis and results

Given that the outcome variable in the analyses represents counts of events (numbers of software libraries deployed), a mixed-effects Poisson regression model (generalized linear mixed model with a Poisson distribution and a log link function) is used to test the research hypotheses. The model includes the variables of deployment context (web or mobile), library ownership (community, proprietary, or big-tech), company scale (Upscale, Future Fifty, or Future Fifty Alumni), and their interactions (full factorial) as fixed effects; company as a random effect; and the number of software libraries as the outcome variable.

The variable of scale is expected to be positively correlated with both company size (number of employees) and age (number of years). However, scale is expected to be more strongly correlated with size than with age because some companies are able to scale up over time while others are not, but only those that scale up also increase in size. In a recent analysis of the phenomenon of blitzscaling, Kuratko et al. (2020) noted that the size of a company grows quickly through this process of aggressive scaling.

Table 6. Results for mixed-effects Poisson regression (web-only companies).

Fixed effect	*F*	*df*1	*df*2	*p*
Ownership	4.909	2	176	0.008
Scale	0.013	2	176	0.987
Ownership × Scale	0.348	4	176	0.845
Age	1.548	1	176	0.215

Note. $n = 186$ (three library deployment values for each of the 62 web-only companies). The outcome variable is the number of software libraries. Only fixed effects are presented. The model includes a random effect for company, which is statistically significant ($p < 0.01$). Information criterion: Akaike corrected is 383.076 and Bayesian is 386.223.

For instance, Amazon grew from 151 employees to 7600 between 1996 and 1999, and LinkedIn grew from 370 to 3458 employees between 2008 and 2012 (Sullivan, 2016). Accordingly, Hoffman and Yeh (2018) structured the main stages of blitzscaling into five aggregate categories by number of employees. Our data confirm these expectations about the relationships of scale with size and age. While scale has a relatively strong relationship with size ($\chi^2_8 = 59.872$, $p < 0.001$), it has a relatively moderate relationship with age ($r = 0.305$, $p < 0.01$). Consequently, age is included in the model to control for time-related variance in the outcome variable, whereas size is excluded for reasons of multicollinearity. Interestingly, there is no statistically significant difference ($t = 0.133$, $p = 0.895$) between the mean age of web-only companies (7.79 years) and the mean age of web-mobile companies (7.91 years), indicating that context multiplicity is independent of company age.

As discussed above, 62 companies have no mobile apps and, consequently, no library deployment in the mobile context (mobile-community, mobile-proprietary, and mobile-big-tech library counts are zero). It is unreasonable to include data on these companies in an effort to fit a model with the context as a variable. Therefore, the full model described above is used to analyse the subgroup of 45 web-mobile companies (those that deploy libraries in both contexts), and a partial model, which excludes the context variable and its interactions with other variables, is used to analyse the subgroup of 62 web-only companies (those that deploy libraries only in the web context). Importantly, as can be inferred from Table 5, because the full model is estimated with 270 observations (six library deployment values for each of the 45 web-mobile companies), and the partial model is estimated with 186 observations (three library deployment values for each of the 62 web-only companies), a lack of statistical power is not a likely explanation for the absence of statistically significant effects.

Web-only companies. The results obtained for the partial model for the 62 web-only companies are presented in Table 6. The estimated mean values are presented in Table 7 and depicted in Figure 1 (two-way interaction of ownership and scale). These results provide support for a statistically significant effect of ownership ($F = 4.909$, $p < 0.01$), according to which web-only companies deploy more proprietary libraries than community libraries ($t = 2.298$, $p < 0.05$) and big-tech libraries ($t = 2.866$, $p < 0.01$). These differences are relatively consistent across all three scale levels, implying that scale and its interaction with ownership do not significantly influence the number of libraries deployed by web-only companies.

The results suggest that web-only companies deploy software libraries representing all ownership types (mean values of the numbers of community, proprietary, and big-tech libraries are significantly different from zero), thereby providing support for our baseline expectation about their coexistence. Furthermore, the results show that web-only companies demonstrate relatively stable deployment of big-tech libraries across scale levels (there are no significant differences among the numbers of big-tech libraries at the three scale levels), providing support for Hypothesis 1. Finally, the results indicate that Hypothesis 2 is not supported for web-only companies. However, while these companies demonstrate no preference for community libraries over proprietary libraries at any scale level (the numbers of community libraries are consistently lower than those of proprietary libraries), Figure 1 suggests that the gap between the deployment of proprietary and community libraries is larger for large-scale companies, consistent with the reasoning underlying Hypothesis 2. Given that web-only companies operate in a single context, Hypothesis 3 cannot be tested with this subgroup alone.

Web-mobile companies. The results obtained for the full model (including the context variable and its interactions with other variables) for the 45 web-mobile companies are presented in Table 8. The estimated mean values are presented in Table 9 and depicted in Figure 2 (two-way interaction of context and ownership) and Figure 3 (three-way interaction of context, ownership, and scale). The results provide support for five statistically significant effects. First, the context has a significant effect on the number of libraries ($F = 5.460$, $p < 0.05$), according to which web-mobile companies deploy more libraries in the web context than in the mobile context. Second, ownership has a significant effect on the number of libraries ($F = 44.813$, $p < 0.001$). While a significant ownership effect is also observed for web-only companies, web-mobile companies deploy more proprietary libraries than big-tech libraries ($t = 4.621$, $p < 0.001$) and more big-tech libraries than community libraries ($t = 5.264$, $p < 0.001$) (deployment of community and big-tech libraries is similar among web-only companies). Third, as depicted in Figure 2 (companies are integrated across scale levels), the two-way interaction of context and ownership is significant ($F = 23.751$, $p < 0.001$), indicating that ownership preferences are different between the two contexts, even within companies. Importantly, the

Table 7. Estimated mean values (web-only companies).

Scale	Ownership			Total
	Community	Proprietary	Big-tech	
Upscale	2.778 (0.454)	3.212 (0.504)	2.300 (0.398)	2.738 (0.351)
Future Fifty	2.594 (0.368)	3.251 (0.430)	2.627 (0.371)	2.809 (0.299)
Future Fifty Alumni	2.431 (0.556)	3.605 (0.731)	2.347 (0.543)	2.740 (0.460)
Total	2.597 (0.269)	3.351 (0.317)	2.421 (0.256)	2.762 (0.213)

Note. $n = 186$. Estimated mean values of number of libraries are presented (with standard errors in parentheses).

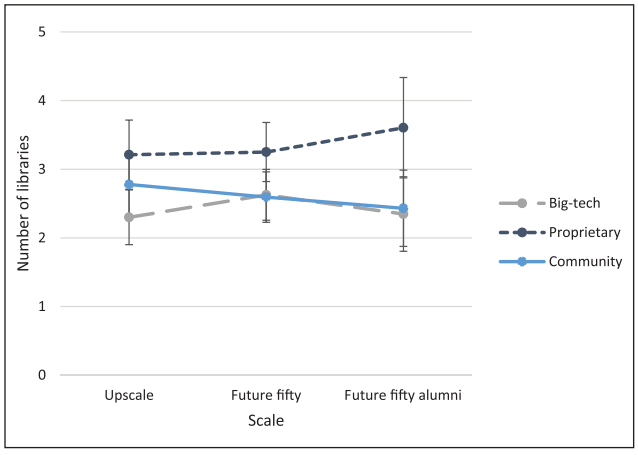

Figure 1. Estimated mean values and standard errors of number of libraries by scale and ownership (web-only companies).

Table 8. Results for mixed-effects Poisson regression (web-mobile companies).

Fixed effect	*F*	*df*1	*df*2	*p*
Context	5.460	1	251	0.020
Ownership	44.813	2	251	0.000
Scale	0.258	2	251	0.773
Context × Ownership	23.751	2	251	0.000
Context × Scale	9.005	2	251	0.000
Ownership × Scale	0.443	4	251	0.777
Context × Ownership × Scale	2.642	4	251	0.034
Age	0.462	1	251	0.497

Note. $n = 270$ (six library deployment values for each of the 45 web-mobile companies). The outcome variable is the number of software libraries. Only fixed effects are presented. The model includes a random effect for company, which is statistically significant ($p < 0.001$). Information criterion: Akaike corrected is 696.096 and Bayesian is 699.606.

ownership preferences of web-mobile companies in the web context are similar to those of web-only companies – more proprietary libraries are deployed than community libraries ($t = 2.209$, $p < 0.05$) and big-tech libraries ($t = 2.163$, $p < 0.05$) – suggesting that library deployment in the web context is independent of whether the company utilizes the mobile context or not. In the mobile context, in addition to the deployment of more proprietary libraries than big-tech libraries ($t = 4.527$, $p < 0.001$), web-mobile companies also deploy more big-tech libraries than community libraries ($t = 6.173$, $p < 0.001$). Fourth, the two-way interaction of context and scale is significant ($F = 9.005$, $p < 0.001$), as only upscale companies deploy more libraries in the web context than in the mobile context ($t = 3.489$, $p < 0.001$). Larger companies demonstrate a more balanced library deployment in web and mobile contexts. Fifth, the three-way interaction of context, ownership, and scale is significant ($F = 2.642$, $p < 0.05$). As depicted in Figure 3, web-mobile companies demonstrate markedly different ownership preferences across scale levels in the web and mobile contexts. Whereas the ownership preferences in the web context (more deployment of proprietary libraries than community and big-tech libraries) are only significant in upscale companies, the ownership preferences in the mobile context (more deployment of proprietary libraries than big-tech libraries and more deployment of big-tech libraries than community libraries) are the smallest in upscale companies (no significant difference in the deployment of proprietary and big-tech libraries). Evidently, over scale levels, ownership preferences converge in the web context and diverge in the mobile context.

The results suggest that, similar to web-only companies, web-mobile companies also deploy libraries that represent all ownership types (mean values of the numbers of community, proprietary, and big-tech libraries are significantly different from zero in each context and in total), providing support for our baseline expectation about their coexistence. Also, similar to web-only companies, web-mobile companies demonstrate relatively stable deployment of big-tech libraries across scale levels (there are no significant differences among the numbers of big-tech libraries at the three scale levels), providing support for Hypothesis 1. Consistent with the results for web-only companies, Hypothesis 2 is not supported for web-mobile companies, as there is no preference for community libraries over proprietary libraries at any scale level. Finally, the results for web-mobile companies provide support for Hypothesis 3. As can be clearly seen in Figure 2, which depicts the significant two-way interaction between context and ownership, companies that operate in both web and mobile contexts demonstrate a stronger preference for proprietary

Table 9. Estimated mean values (web-mobile companies).

Scale	Context	Ownership			Total
		Community	Proprietary	Big-tech	
Upscale	Mobile	0.420 (0.222)	2.733 (0.714)	1.997 (0.574)	1.319 (0.343)
	Web	2.733 (0.714)	5.466 (1.211)	2.207 (0.614)	3.206 (0.654)
	Total	1.072 (0.342)	3.865 (0.813)	2.100 (0.492)	2.056 (0.423)
Future Fifty	Mobile	0.744 (0.203)	6.374 (0.876)	3.024 (0.497)	2.430 (0.352)
	Web	2.280 (0.408)	2.373 (0.419)	2.699 (0.458)	2.444 (0.327)
	Total	1.303 (0.234)	3.889 (0.526)	2.857 (0.401)	2.437 (0.304)
Future Fifty Alumni	Mobile	0.846 (0.242)	6.938 (1.064)	3.272 (0.591)	2.678 (0.428)
	Web	2.031 (0.422)	2.313 (0.461)	2.143 (0.438)	2.159 (0.335)
	Total	1.311 (0.259)	4.006 (0.614)	2.648 (0.429)	2.405 (0.341)
Total	Mobile	0.642 (0.141)	4.944 (0.538)	2.703 (0.335)	2.047 (0.227)
	Web	2.330 (0.290)	3.107 (0.353)	2.337 (0.296)	2.567 (0.241)
	Total	1.223 (0.168)	3.919 (0.374)	2.514 (0.261)	2.293 (0.208)

Note. $n = 270$. Estimated mean values of number of libraries are presented (with standard errors in parentheses).

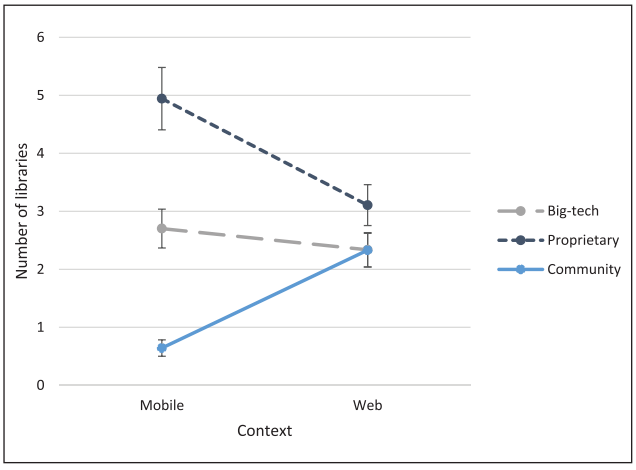

Figure 2. Estimated mean values and standard errors of number of libraries by context and ownership (web-mobile companies).

libraries over community libraries in the mobile context relative to the web context. Compared to the web context, the mobile context is characterized by the deployment of more proprietary libraries ($t = 3.798$, $p < 0.001$) and less community libraries ($t = -5.745$, $p < 0.001$), with no significant difference in the deployment of big-tech libraries. Table 10 provides a summary of the results of hypothesis testing.

Discussion

This work studies two major software development contexts that have evolved along two different paths of infrastructure ownership and finds in both a cohabitation of community, proprietary, and big-tech infrastructural components. Therefore, the emerging landscape is neither the community-based utopia of early open-source research (Benkler, 1998; Hess and Ostrom, 2007; Lessig, 1999) nor the dystopia of the recent digital dominance literature (Benkler et al., 2018; Khan, 2018; Moore and Tambini, 2018; Zuboff, 2015).

The infrastructural landscape we observe provides novel insights into the role of infrastructure ownership. The first is the stability of big-tech dominance, as we see that the deployment of big-tech libraries is independent of company scale and software development context. These findings suggest that big-tech libraries have become foundational infrastructure, while libraries from proprietary vendors and communities serve as additional, optional resources. An interesting avenue for future research is to examine these foundational and complementary roles of software libraries, contingent on their ownership. A second insight is the lower deployment of libraries in the mobile context than in the web context. The operating systems and SDKs provided by Apple and Google are attractors (Braa et al., 2007) that offer much of the functionality needed by downstream providers. Because the web is based on open standards, designed to provide minimal communication and content presentation functionality (Berners-Lee et al., 1992; Siegele, 2018), libraries or extensions are demanded in the web context. Finally, a surprising finding is that smaller companies do not prefer community libraries over proprietary libraries. While the open-source argument about autonomy suggests that the freedom to access and change the open-source code itself represents an advantage over proprietary software (Lee and Cole, 2003; Shaikh and Vaast, 2016; Zeitlyn, 2003), other benefits of proprietary libraries presumably outweigh this benefit of community libraries for smaller companies.

Contributions and implications

This study contributes to the literature on digital infrastructure in several ways. First, we extend previous research on

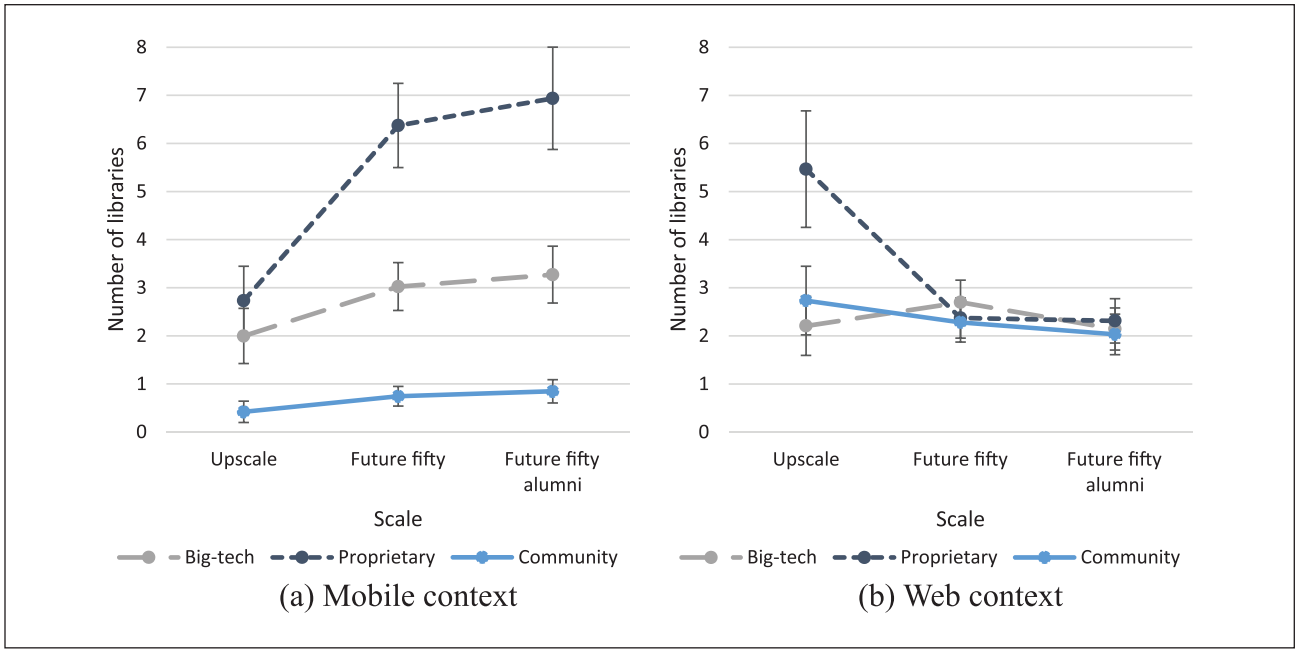

Figure 3. Estimated mean values and standard errors of number of libraries by scale and ownership in either (a) mobile or (b) web context (web-mobile companies).

Table 10. Summary of hypothesis testing results.

Hypothesis		Web-only companies	Web-mobile companies
Hypothesis 1	Companies of different scales similarly deploy big-tech libraries.	Supported	Supported
Hypothesis 2	Small-scale companies demonstrate a stronger preference for community libraries over proprietary libraries relative to large-scale companies.	Not supported	Not supported
Hypothesis 3	Companies that operate in both web and mobile contexts demonstrate a preference for proprietary libraries over community libraries in the mobile context relative to the web context.	N/A	Supported

digital infrastructure (Constantinides et al., 2018; De Reuver et al., 2018; Tilson et al., 2010) by considering software libraries as infrastructure, and we extend previous research on software reuse (Flath et al., 2017; Haefliger et al., 2008) by analysing its infrastructural aspect. We further develop the conceptualization of digital infrastructure as boundary resources (Eaton et al., 2015; Ghazawneh and Henfridsson, 2010, 2013) and argue that software libraries address different trade-offs depending on whether they are owned by communities, proprietary vendors, or big-tech providers. Second, we extend the traditional distinction between public and private resources by considering resources owned by dominant digital providers as a third ownership type. We demonstrate that big-tech libraries are deployed differently than community and proprietary libraries. A third contribution of this study is that it sheds light on the revealed preferences of infrastructure users, such as the stable preference for big-tech libraries, despite their lower autonomy and higher cost. Therefore, this work also contributes to the OSS literature, which has focused on software development (Ågerfalk et al., 2015), but has paid relatively little attention to software deployment. A final contribution is the balanced portrayal of the current landscape of digital infrastructure that embodies a new mix of public and private resources. The evidence emerging from our analysis complements research on the limits of private-collective innovation (Von Hippel and Von Krogh, 2003) and questions some of the predictions of open-source research about the freedom to innovate (Benkler, 1998; Hess and Ostrom, 2007; Lessig, 1999).

From a practical standpoint, development managers may reason that it is beneficial to deploy community and big-tech libraries as needed, particularly when big-tech libraries are free and there is a low likelihood of future fees or alternative mechanisms of control by big-tech providers. While this reasoning holds in the current web and mobile contexts, it was not applicable earlier because big-tech vendors did not provide free uncontrolled functionality. The

high deployment of proprietary libraries in the mobile context is probably an outcome of the exponential growth of mobile apps. The rapid evolution of this market may require development managers to react quickly and, consequently, pay for infrastructure. From a public policy standpoint, the current debate on big-tech is important and requires a better understanding of the implications of dominance on the way companies deploy a variety of infrastructural components.

Limitations and future research

From a conceptual standpoint, a limitation of this study is its focus on the digital infrastructure literature and the limited attention given to several literature streams that may enrich its conceptual framing. These streams include the literature on platforms (Gawer and Cusumano, 2002; Parker and Van Alstyne, 2005), business and entrepreneurial ecosystems (Autio et al., 2018; Iansiti and Levien, 2004; Jacobides et al., 2018), and communities of practice and online communities (Faraj et al., 2016; Paasivaara and Lassenius, 2014; Wenger, 1998). Although these literature streams have already influenced digital infrastructure research, such as the consideration of the duality of control and autonomy on the basis of the platform literature (Eaton et al., 2015; Ghazawneh and Henfridsson, 2013), additional significant advancements can be made by allowing these literature streams to further enrich digital infrastructure research.

From an empirical standpoint, a limitation of this study is its limited external validity. As discussed above regarding the research setting, our empirical analysis is based on the observation of software library deployment by 107 companies that participated in a high-profile digital innovation cluster. Consequently, caution should be exercised in generalizing the findings of this study to the entire population of entrepreneurial companies at large. Notwithstanding this limitation, the lack of quantitative evidence in the digital infrastructure literature requires researchers at this point in time to focus on the empirical identification of the variables of interest and their measures. Only after key variables and valid measures are identified can more thorough research take into account considerations of representativeness and control by observing multiple contextual variables in cross-sectional settings or seek to observe the usefulness of low-level components of digital infrastructure rather than their deployment (i.e. observe their utility rather than their demand).

A second, related empirical limitation is that we observe the quantity of libraries but not their functionality and quality. The models we analyse include a measure of the number of libraries but no measures of their functionality, implying that a library that provides rich functionality with many services is counted the same as a lean, single-service library. While our empirical approach relies on the assumption that variance in library functionality is less of an issue at the aggregate level, future research should incorporate measures of library functionality to provide a more complete account of differences in the deployment of community, proprietary, and big-tech libraries.

A third empirical limitation is related to the digital tools used to measure deployment. Although Wappalyzer and AppBrain are well-known tools in the relevant communities, and similar techniques have been reported in the literature (Greenstein and Nagle, 2014; Martin et al., 2017; Spinellis and Giannikas, 2012), the use of two independent tools with different baseline repositories and collection techniques may raise concerns of potential confounding. Given that our analysis focuses on within-context comparisons more than between-context comparisons (only investigated in Hypothesis 3), this potential confounding is not detrimental to our findings. While further research on software library deployment in a larger variety of contexts is called for, a major challenge for such research will be the development of more uniform data collection tools and techniques.

Additional directions for future research include further extension of the boundary resource concept to a wider collection of artefacts, including programming languages and design patterns as abstract resources as well as SDKs, APIs, libraries, and code snippets (Baltes and Diehl, 2019) as concrete resources. Further research is also required to better understand the mechanisms that allow for low-cost experimentation with big-tech and proprietary infrastructure. Such mechanisms should explain why we observe variance in the deployment of libraries owned by smaller providers but no variance in the deployment of libraries owned by dominant providers. A third direction for future research is related to the finding that the deployment of libraries reflects their context. While Hypothesis 3 highlights differences between contexts, future research should address the factors that determine how the context shapes infrastructural relations (Tilson et al., 2010). These relations emerge as sociotechnical practices between innovators and users, upstream and downstream businesses, keystone and niche players, platform leaders and followers, and developer communities and their joiners. Little is known about how contexts and their specific boundary resources influence these relationships (Germonprez et al., 2017; Von Krogh et al., 2003). Furthermore, the organization of innovation requires digital infrastructure and, in turn, the new and innovative becomes infrastructure over time. The mechanisms underlying these processes may be more complex than currently acknowledged, and future research in these directions is needed to advance our understanding of digital infrastructure.

Conclusion

With the aim of understanding how the ownership of boundary resources affects their use, this study investigates

how the ownership of software libraries affects their deployment by 107 start-up companies in London. The analysis confirms the coexistence of community, proprietary, and big-tech libraries in both web and mobile contexts. We show that library deployment, in particular the deployment of big-tech libraries, is relatively stable across different company scales and that the preference for proprietary libraries over community libraries is stronger in the mobile context than in the web context. Overall, our findings suggest that libraries owned by different types of owners are able to cater to different needs of deploying companies. As the societal role of digital infrastructure continues to evolve, and as it becomes increasingly difficult to distinguish between private and public resources, this work depicts an infrastructural landscape that is neither a community-based utopia nor a digital dominance dystopia.

Acknowledgement

The authors wish to thank audiences at the Academy of Management Big Data conference in Surrey and the Barclays data science group for helpful comments along the way as well as, specifically, Charles Baden-Fuller, Annabelle Gawer, Hans Berends, Carsten Sorensen, Phillip Tuertscher, and our editor Timothy Coltman and a supportive review team at JIT.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: We would like to acknowledge the grant from the U.K. Engineering and Physical Sciences Research Council (EPSRC) EP/K039695/1 that supported this research at Cass Business School.

ORCID iD

Lior Fink https://orcid.org/0000-0001-5165-5259

References

Ågerfalk PJ, Fitzgerald B and Stol K-J (2015) *Software Sourcing in the Age of Open: Leveraging the Unknown Workforce*. Cham: Springer.

Alexy O, Henkel J and Wallin MW (2013) From closed to open: Job role changes, individual predispositions, and the adoption of commercial open source software development. *Research Policy* 42(8): 1325–1340.

Autio E, Nambisan S, Thomas LDW, et al. (2018) Digital affordances, spatial affordances, and the genesis of entrepreneurial ecosystems. *Strategic Entrepreneurship Journal* 12(1): 72–95.

Baltes S and Diehl S (2019) Usage and attribution of Stack Overflow code snippets in GitHub projects. *Empirical Software Engineering* 24(3): 1259–1295.

Barns BH and Bollinger TB (1991) Making reuse cost-effective. *IEEE Software* 8(1): 13–24.

Barwise P and Watkins L (2018) The evolution of digital dominance: How and why we got to GAFA. In: Moore M and Tambini D (eds) *Digital Dominance: The Power of Google, Amazon, Facebook, and Apple*. Oxford: Oxford University Press, pp. 21–49.

Benkler Y (1998) Overcoming agoraphobia: Building the commons of the digitally networked environment. *Harvard Journal of Law & Technology* 11(2): 1–113.

Benkler Y (2013) Commons and growth: The essential role of open commons in market economies. *The University of Chicago Law Review* 80(3): 1499–1555.

Benkler Y (2016) Peer production and cooperation. In: Bauer JM and Latzer M (eds) *Handbook on the Economics of the Internet*. Northampton, MA: Edward Elgar Publishing, pp. 91–119.

Benkler Y, Faris R and Roberts H (2018) *Network Propaganda: Manipulation, Disinformation, and Radicalization in American Politics*. Oxford: Oxford University Press.

Berners-Lee T, Cailliau R, Groff JF, et al. (1992) World-Wide Web: The information universe. *Internet Research* 2(1): 52–58.

Bouwman CHS, Hu S and Johnson SA (2018) Differential bank behaviors around the Dodd–Frank Act size thresholds. *Journal of Financial Intermediation* 34: 47–57.

Braa J, Hanseth O, Heywood A, et al. (2007) Developing health information systems in developing countries: The flexible standards strategy. *MIS Quarterly* 31(2): 381–402.

Brooks FP (1975) *The Mythical Man Month: Essays on Software Engineering*. Reading, MA: Addison-Wesley.

Constantinides P, Henfridsson O and Parker GG (2018) Platforms and infrastructures in the digital age. *Information Systems Research* 29(2): 381–400.

Crémer J, de Montjoye Y-A and Schweitzer H (2019) *Competition Policy for the Digital Era*. Brussels: European Commission. Available at: http://ec.europa.eu/competition/publications/reports/kd0419345enn.pdf (accessed 2 May 2019)

Cusumano MA (2012) Platforms versus products: Observations from the literature and history. In: Kahl S, Silverman B and Cusumano MA (eds) *History and Strategy*. Bingley: Emerald Group Publishing Limited, pp. 35–67.

Cusumano MA and Selby RW (1997) How Microsoft builds software. *Communications of the ACM* 40(6): 53–61.

Cusumano MA and Selby RW (1998) *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People*. New York: Simon & Schuster.

De Reuver M, Sørensen C and Basole RC (2018) The digital platform: A research agenda. *Journal of Information Technology* 33(2): 124–135.

Eaton B, Elaluf-Calderwood S, Sørensen C, et al. (2015) Distributed tuning of boundary resources: The case of Apple's iOS service system. *MIS Quarterly* 39(1): 217–243.

European Commission (2016) Antitrust: Commission sends statement of objections to Google on Android operating system

and applications – Factsheet. Available at: http://europa.eu/rapid/press-release_MEMO-16-1484_en.htm (accessed on 19 February 2018)

European Commission (2018) Antitrust: Commission fines Google €4.34 billion for illegal practices regarding Android mobile devices to strengthen dominance of Google's search engine. Available at: http://europa.eu/rapid/press-release_IP-18-4581_en.htm (accessed on 29 July 2018)

Faraj S, von Krogh G, Monteiro E, et al. (2016) Special section introduction – Online community as space for knowledge flows. *Information Systems Research* 27(4): 668–684.

Feldman MP and Kogler DF (2010) Stylized facts in the geography of innovation. In: Hall BH and Rosenberg N (eds) *Handbook of the Economics of Innovation*, vol. 1. Oxford: North-Holland, pp. 381–410.

Fitzgerald B (2006) The transformation of open source software. *MIS Quarterly* 30(3): 587–598.

Flath CM, Friesike S, Wirth M, et al. (2017) Copy, transform, combine: Exploring the remix as a form of innovation. *Journal of Information Technology* 32(4): 306–325.

Gawer A and Cusumano MA (2002) *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*. Boston, MA: Harvard Business School Press.

Germonprez M, Kendall JE, Kendall KE, et al. (2017) A theory of responsive design: A field study of corporate engagement with open source communities. *Information Systems Research* 28(1): 64–83.

Ghazawneh A and Henfridsson O (2010) Governing third-party development through platform boundary resources. In: *Proceedings of the thirty first international conference on information systems*, Saint Louis, MO, 12–15 December, Paper 48.

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

Greenstein S and Nagle F (2014) Digital dark matter and the economic contribution of Apache. *Research Policy* 43(4): 623–631.

Haefliger S, von Krogh G and Spaeth S (2008) Code reuse in open source software. *Management Science* 54(1): 180–193.

Henfridsson O and Bygstad B (2013) The generative mechanisms of digital infrastructure evolution. *MIS Quarterly* 37(3): 907–931.

Hess C and Ostrom E (2007) Introduction: An overview of the knowledge commons. In: Hess C and Ostrom E (eds) *Understanding Knowledge as a Commons: From Theory to Practice*. Cambridge, MA: MIT Press, pp. 3–26.

Hoffman R and Yeh C (2018) *Blitzscaling: The Lightning-Fast Path to Building Massively Valuable Companies*. New York: Random House.

Iansiti M and Levien R (2004) *The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability*. Boston, MA: Harvard Business School Press.

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

Karasti H and Blomberg J (2018) Studying infrastructuring ethnographically. *Computer Supported Cooperative Work* 27(2): 233–265.

Khan ML (2018) Amazon – An infrastructure service and its challenge to current antitrust law. In: Moore M and Tambini D (eds) *Digital Dominance: The Power of Google, Amazon, Facebook, and Apple*. Oxford: Oxford University Press, pp. 98–132.

Kim Y and Stohr EA (1998) Software reuse: Survey and research directions. *Journal of Management Information Systems* 14(4): 113–147.

Kogut B and Metiu A (2001) Open-source software development and distributed innovation. *Oxford Review of Economic Policy* 17(2): 248–264.

Kuratko DF, Holt HL and Neubert E (2020) Blitzscaling: The good, the bad, and the ugly. *Business Horizons* 63(1): 109–119.

Kuwatly I, Sraj M, Masri ZA, et al. (2004) A dynamic honeypot design for intrusion detection. In: *The IEEE/ACS international conference on pervasive services*, Beirut, Lebanon, 19–23 July 2004, pp. 95–104. New York: IEEE.

Lee GK and Cole RE (2003) From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization Science* 14(6): 633–649.

Lerner J and Schankerman M (2010) *The Comingled Code: Open Source and Economic Development*. Cambridge, MA: MIT Press.

Lessig L (1999) *Code and Other Laws of Cyberspace*. New York: Basic Books.

Levy S (1994) *Insanely Great: The Life and Times of Macintosh, the Computer That Changed Everything*. New York: Viking.

Martin W, Sarro F, Jia Y, et al. (2017) A survey of app store analysis for software engineering. *IEEE Transactions on Software Engineering* 43(9): 817–847.

Moody G (2001) *Rebel Code: Linux and the Open Source Revolution*. Cambridge, MA: Perseus.

Moore M and Tambini D (2018) *Digital Dominance: The Power of Google, Amazon, Facebook, and Apple*. Oxford: Oxford University Press.

O'Mahony S (2003) Guarding the commons: How community managed software projects protect their work. *Research Policy* 32(7): 1179–1198.

Osterwalder A and Pigneur Y (2010) *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Hoboken, NJ: John Wiley & Sons.

Ostrom E (2015) *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge: Cambridge University Press.

Paasivaara M and Lassenius C (2014) Communities of practice in a large distributed agile software development organization – Case Ericsson. *Information and Software Technology* 56(12): 1556–1577.

Parker GG and Van Alstyne MW (2005) Two-sided network effects: A theory of information product design. *Management Science* 51(10): 1494–1504.

Pon B, Seppälä T and Kenney M (2014) Android and the demise of operating system-based power: Firm strategy and platform control in the post-PC world. *Telecommunications Policy* 38(11): 979–991.

Ribes D and Lee CP (2010) Sociotechnical studies of cyberinfrastructure and e-research: Current themes and future trajectories. *Computer Supported Cooperative Work* 19(3–4): 231–244.

Rock D (2019) Engineering value: The returns to technological talent and investments in artificial intelligence. Available at: https://ssrn.com/abstract=3427412 (accessed on 8 January 2020)

Rolland KH, Mathiassen L and Rai A (2018) Managing digital platforms in user organizations: The interactions between digital options and digital debt. *Information Systems Research* 29(2): 419–443.

Shaikh M and Vaast E (2016) Folding and unfolding: Balancing openness and transparency in open source communities. *Information Systems Research* 27(4): 813–833.

Siegele L (2018) How to fix what has gone wrong with the Internet, special report. *The Economist*. Available at: https://www.economist.com/special-report/2018/06/28/how-to-fix-what-has-gone-wrong-with-the-internet (accessed on August 1, 2018)

Sommerville I (2000) *Software Engineering*. Reading, MA: Addison-Wesley.

Spinellis D and Giannikas V (2012) Organizational adoption of open source software. *Journal of Systems and Software* 85(3): 666–682.

Stallman RM (1985) The GNU manifesto. Available at: http://www.gnu.org/gnu/manifesto.html (accessed on January 8, 2020)

Star SL and Ruhleder K (1996) Steps toward an ecology of infrastructure: Design and access for large information spaces. *Information Systems Research* 7(1): 111–134.

Stewart KJ and Gosain S (2006) The impact of ideology on effectiveness in open source software development teams. *MIS Quarterly* 30(2): 291–314.

Sullivan T (2016) Blitzscaling. *Harvard Business Review* 94(4): 45–50.

Tech City UK (2017) Tech Nation 2017. Available at: https://technation.techcityuk.com/cluster/london/ (accessed on 26 February 2018)

Tilson D, Lyytinen K and Sørensen C (2010) Digital infrastructures: The missing IS research agenda. *Information Systems Research* 21(4): 748–759.

Tiwana A, Konsynski B and Bush AA (2010) Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research* 21(4): 675–687.

U.S. v. Microsoft Corporation, 231 F. Supp. 2d 144 (D.D.C. 2002). United States District Court, District of Columbia.

Vaast E and Walsham G (2009) Trans-situated learning: Supporting a network of practice with an information infrastructure. *Information Systems Research* 20(4): 547–564.

Von Hippel E and Von Krogh G (2003) Open source software and the 'private-collective' innovation model: Issues for organization science. *Organization Science* 14(2): 209–223.

Von Krogh G, Spaeth S and Lakhani KR (2003) Community, joining, and specialization in open source software innovation: A case study. *Research Policy* 32(7): 1217–1241.

Warren E (2019) Here's how we can break up big tech. *Medium.com*. Available at: https://medium.com/@teamwarren/heres-how-we-can-break-up-big-tech-9ad9e0da324c (accessed on 2 May 2019)

Wenger E (1998) *Communities of Practice: Learning, Meaning, and Identity*. Cambridge: Cambridge University Press.

Woodard CJ, Ramasubbu N, Tschang FT, et al. (2013) Design capital and design moves: The logic of digital business strategy. *MIS Quarterly* 37(2): 537–564.

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

Yoo Y, Lyytinen K, Boland R, et al. (2010b) The next wave of digital innovation: Opportunities and challenges. Available at: http://ssrn.com/abstract=1622170 (accessed on 16 April 2020)

Zeitlyn D (2003) Gift economies in the development of open source software: Anthropological reflections. *Research Policy* 32(7): 1287–1291.

Zuboff S (2015) Big other: Surveillance capitalism and the prospects of an information civilization. *Journal of Information Technology* 30(1): 75–89.

Zuboff S (2019) *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. London: Profile Books.

Author biographies

Lior Fink is an associate professor of information systems in the Department of Industrial Engineering and Management at Ben-Gurion University of the Negev, Israel. He holds a bachelor's degree in psychology and economics, a master's degree in social-industrial psychology, and a PhD degree in information systems from Tel Aviv University. He held visiting positions at the UCLA Anderson School of Management and the Smith School of Business, University of Maryland, College Park. His research interests focus on economic and strategic aspects of IT development, outsourcing, and use. His articles have been published in numerous information systems journals, including *MIS Quarterly, Journal of the Association for Information Systems, European Journal of Information Systems, Information Systems Journal, Journal of Information Technology*, and *Journal of Strategic Information Systems*. His work has also appeared in economics, psychology, project management, and medical informatics journals.

Jianhua Shao is employed by the Alibaba Group (Europe) as an advanced analytics expert and thought leader in digital transformation, with responsibility for the design of digital solutions for Alibaba's client companies. Previously, he was a data scientist at Barclays Bank, T-Mobile, and China Mobile. He holds a PhD in digital economy and BSc in computer science from Nottingham University, and is a fellow at Cass Business School, City University of London.

Yossi Lichtenstein is a teaching fellow at Queen Mary University of London. He holds a bachelor's and master's degrees in computer science and a PhD degree in Information Systems from Tel

Aviv University. His articles have been published in numerous journals, including *MIS Quarterly, Communications of the ACM* and *IEEE Software.*

Stefan Haefliger is a professor of strategic management and innovation at Cass Business School, City University of London and a faculty member at the Department of Management, Technology, and Economics, ETH Zurich. He received a master's degree in financial economics and a PhD in management science from the University of St. Gallen in Switzerland. Stefan held visiting positions at Università degli Studi di Trento, MIT, Hitotsubashi and Politecnico di Milano. In his research and teaching, he focuses on co-creation strategies as well as knowledge reuse, creation, and design in innovation processes. His research has appeared in numerous journals, including *Management Science, Research Policy* and *MIS Quarterly.*