Proprietary or open source software? Winner-takes-all competition, partial adoption and efficiency

Eric Darmon, Thomas Le Texier and Dominique Torre

Electronic version

URL: <https://journals.openedition.org/rei/5223>

DOI: 10.4000/rei.5223

ISSN: 1773-0198

Publisher

De Boeck Supérieur

Printed version

Date of publication: 15 December 2011

Number of pages: 109-140

ISSN: 0154-3229

Electronic reference

Eric Darmon, Thomas Le Texier and Dominique Torre, "Proprietary or open source software? Winner-takes-all competition, partial adoption and efficiency", *Revue d'économie industrielle* [Online], 136 | 4ème trimestre 2011, Online since 15 December 2013, connection on 02 June 2022. URL: <http://journals.openedition.org/rei/5223> ; DOI: <https://doi.org/10.4000/rei.5223>

Éric DARMON and Thomas LE TEXIER
Université de Rennes 1, CREM – UMR 6211 CNRS

Dominique TORRE
Université de Nice – Sophia Antipolis, GREDEG – UMR 6227 CNRS

PROPRIETARY OR OPEN SOURCE SOFTWARE ? WINNER-TAKES-ALL COMPETITION, PARTIAL ADOPTION AND EFFICIENCY

Key words : Open Source Software, Proprietary Software, Competition, Increasing Returns, Community of Developers.

Mots-clés : Logiciel open-source, logiciel propriétaire, concurrence, communautés de développeurs, externalités de compatibilité et de développement.

I. — INTRODUCTION

The emergence in the software industry of an alternative production model based on free/libre activities has generated new and original competition outcomes. The economic analysis of these outcomes reveals two major trends.

First, the emergence of original cooperation patterns between open source communities and commercial editors of proprietary software, which have led to « hybrid » business models and several studies show how software firms are leveraging open source projects for commercial purposes (Hertel *et al.*, 2003 ; Henkel, 2004 ; Dahlander and Magnusson, 2005 ; von Krogh and von Hippel, 2006 ; Harhoff and Mayrhofer, 2008 ; Rolandsson *et al.*, 2011). Within this fra-

Acknowledgments: The authors would like to acknowledge Stefano Comino, Jean-Michel Dalle, Frédéric Deroian, Cynthia Little, Christian Longhi, Edward Lorenz, Thierry Pénard, Cristina Rossi, Alexandra Rufini, Francesco Rullani, Philipp Schröder, Yossi Spiegel, David Versailles, Jean-Benoît Zimmerman and an anonymous referee of *La Revue d'Économie Industrielle* for their comments and suggestions on preliminary versions of this paper. Only the authors are responsible of the remaining errors and imperfections of the article.

mework, cooperative participative patterns can be beneficial to both community-based and commercial organizations provided suitable governance mechanisms have been defined. There are numerous examples of commercial-community alignments in the literature (Bonaccorsi *et al.*, 2006; Dahlander and Wallin, 2006; Dahlander, 2007; Dahlander and Magnusson, 2008; Ghosh *et al.*, 2008; West and Lakhani, 2008; Riehle, 2009; Stam, 2009).

Second, the open source software development model is allowing new players to compete in the software market. Competition in the software market is governed by different parameters (such as the quality, functionalities, and cost of proprietary solutions, or the size of open source communities) and the co-existence of proprietary and open software solutions is likely to depend on the environment in which the software is produced (*e.g.*, type and magnitude of network externalities on users' and/or developers' side). For instance, some authors find that open source software is likely to overtake proprietary solutions for both informational (Kuan, 2002) and quality reasons (Bessen, 2002). Others find that firms are more likely to persist in the competitive market than open source projects (Casadesus-Masanell and Ghemawat, 2006). However, many studies show that both types of software production activities are likely to co-exist in the market. For instance, Bonaccorsi and Rossi (2003), using an agent-based model, show that proprietary and libre software will co-exist when network effects are moderate and Johnson (2002) stresses that the success of open source over proprietary solutions applies only when users invest a certain level of effort to contributing open source code. Similarly, in a mixed duopoly model in which producers' objective functions differ, Economides and Katsamakas (2006) find that competition outcomes may lead to a shared market situation. They analyse a two-sided competition model involving an open source and a proprietary platform. Dalle and Jullien (2000; 2003) present a dynamic simulation of the diffusion of Linux that considers local and global network effects. They point to the crucial role of early adopters for the success or failure of open source projects and underline the large impact of compatibility and governance models on competition outcomes. Thus, the co-existence of both types of software in the market is a possible competition outcome.

However some theoretical studies find that competition outcomes are less clear and are dependent on the way commercial firms react to the libre activity. Bitzer (2004) focuses on the impact of product-based heterogeneity, showing that incumbents (*i.e.*, proprietary software firms) can remain profitable by setting higher prices. Following the job signalling hypothesis formulated by Lerner and Tirole (2002), Mustonen (2003) focuses on the role in competition of the wages paid by proprietary firms. He shows that commercial firms can crowd out the diffusion of the open source software solution by offering higher wages, and that both types of software can co-exist in the market. Lanzi (2009) finds that, in a duopoly setting, competition outcomes depend on the level of complexity of the open source software solution, such that open source may dominate if learning costs are sufficiently low but a shared-market outcome is

more likely when learning costs are high. Välimäki and Oksanen (2005) focus on the role of switching costs in software adoption patterns and highlight various competition outcomes. Leoncini *et al.* (2008) introduce different types of innovation patterns and interoperability features in a dynamic model and find that a perfect interoperability innovation pattern is likely to lead to a shared-market outcome whereas the results are more ambiguous for other innovation patterns.

This article analyses adoption dynamics in the market for software when a proprietary software firm competes with an open source software community. We make several contributions to the literature. First, we introduce the role of expectations about adoption patterns. Most of the existing literature do not explicitly consider this issue and focus on particular type of equilibrium. We here analyze the whole set of (Nash) equilibria coming from the introduction of open source software. Second, we characterize software as an experience good. Such a feature is critical when dealing with adoption issues, since the quality of the software cannot in this case be perfectly assessed *ex ante*. Third, we consider adoption patterns taking explicit account of the coexistence of heterogeneous users (*i.e.*, users with different programming abilities and distinct preferences). We use the typology in Franke and von Hippel (2003) to distinguish *developer-users* and *end-users*. We define a theoretical framework in which the need for specific functionalities not provided by proprietary software are the motivation for development. This framework takes account of differences in licensing costs when proprietary software is substituted by open source software released under GPL (General Public License) terms. We define a theoretical competition game model between a monopolistic firm that produces proprietary software and a community that develops alternative GPL-based open source software. In this framework, we identify the strategies that proprietary software firms may adopt to remain on the market. We also evaluate the welfare generated by the introduction of *libre* solutions in order to measure the social effect of open source software, something that few studies have focused on so far (Schmidt and Schnitzer, 2003; von Engelhardt and Maurer, 2010).

We show that the final levels of adoption of both open source and proprietary software depend on the initial conditions relative to users' expectations. In some settings, winner-takes-all competition may arise, which may lead to the crowding out of one or other of the types of software. Other settings lead to a mixed equilibrium with users distributed between the two types of software according to their abilities to develop or to adopt, and to the software adoption costs. In most cases, the introduction of Open Source Software as a potential competitor leads to multiple equilibria. Such multiple equilibria are imperfectly controlled by the proprietary firm, thus, its strategy can be understood as a balance between a low price-high quality and high price-low quality strategy. By analysing the qualitative properties of winner-takes-all and separating equilibria, we show that the existence of a sole credible open source alternative improves the end user's utility, even if ultimately only the proprietary solu-

tion is adopted. In that respect, open source software may be considered as a potential threat on the software market. From consumers' point of view, such threat drives the market to a more favourable situation. However, we show that, under some circumstances, the diffusion of open source software generates conflicts of interest. In some cases, there is a divide between users and the firm, but also the interests of some users are aligned to those of the firm.

The paper is organized as follows. Section 2 presents the model. Section 3 analyses the Nash equilibrium generated by the model. Section 4 discusses the qualitative properties of this equilibrium and introduces welfare-related issues. Section 5 concludes.

II. — THE MODEL – GENERAL SETTINGS

The model analyses the interactions between a firm producing proprietary software and two categories of users, end-users and developer-users, defined by their ability to contribute to the development of open source software.

We characterize the model as a two-step game. First, the firm defines a price and quality strategy. Second, potential users decide whether to adopt or not.

2.1. End-users and developers

There are two categories of users, namely developers (D , in proportion μ of the total population) and end-users (EU , in proportion $(1 - \mu)$ of the total population whose size is set to 1). End-users and developers differ in two ways.

First, developers are able to develop new functionalities and customize open source software by adding new lines of code, but end-users are not. In other words, since source code is open, developers can develop new functionalities for their own benefit. Once these functionalities are developed, there is no cost in releasing them to the whole community of developers and in giving end-users free access to these functionalities (1). Second, the adoption costs are different for end-users and developers. Developers are generally expert users (*e.g.*, engineers, computer scientists) and are used to adopting new software, while end-users may face high initial adoption costs in choosing new software. Also, it seems reasonable to assume that these costs are generally higher for

- (1) We restrict our analysis to GPL licences. Other types of licences (*e.g.*, the BSD licence) allow any user or developer to commercialize their own software derived from initial software. These other types of licences raise additional incentive problems for developers who may be motivated to « close » the code of their customized software instead of making it freely available to other users. This specific case could be studied as an extension of this article.

open source software than would be incurred in the adoption of a proprietary solution. One reason for this is that open source software is developed primarily in open source projects which often are led by expert users who do not have time to devote extensive tutorials/FAQs, user-friendly interfaces, etc. (2).

For all these reasons, we suppose that developers – as opposed to end-users – do not face adoption costs when adopting open source software. Among the population of end-users, though, some agents will be more likely to adopt open source software more quickly. This varying ability translates into heterogeneous adoption costs, which here are assumed to be uniformly distributed between 0 and $\bar{c}^{EU}$ ($\bar{c}^{EU} > 0$).

2.2. Proprietary software and open source software

The two types of software need to provide both generic and specialized functionalities. Generic functionalities are commonly used by all users; specific functionalities differ according to each user's needs. To depict this heterogeneity, we assume that users are uniformly distributed on a unit circle. Users can choose (i) to adopt proprietary – closed source – software, (ii) to adopt GPL – based open source software, or (iii) not to adopt. In the last situation, we suppose that users can accomplish the same task manually or using a combination of previously acquired software (*i.e.*, at no purchase cost). By convention, this default option leads to a null payoff.

2.3. Strategy of the proprietary firm

We suppose that a single commercial firm produces proprietary software. Because software is a digital good, its reproduction (or variable) costs are negligible (Shapiro and Varian, 1998). We assume them to be null and independent of the number of adopters. The development cost depends on the quality q of the software that we assimilate in the extent of the specialized functionalities covered by the software. By devoting more effort, the firm can develop more specialized functionalities in order to cover more specific needs on the unit circle. Then the quality q is depicted by the number of available specialized functionalities and is expressed as a fraction of the unit circle circumference ($q \in [0,1]$). The quality and its related cost $g(q)$ are controlled by the firm. Since users are uniformly distributed, it is reasonable to suppose that the software editor first develops those functionalities that are less costly. Let us assume, therefore, that

- (2) It could be argued that users may incur an adoption cost even when adopting proprietary software. However, because user-friendliness is an important factor in adoption, commercial firms devote marketing efforts to maximizing the accessibility of their software. Thus, it seems reasonable to distinguish open source from proprietary software by overlooking the adoption costs required to use proprietary software.

development costs increase at an increasing rate with quality q (i.e., $g''(.) > 0$). One simple specification is given by $g(q) = \beta q^2$ ($\beta > 0$).

Let us denote by c_L ($c_L > 0$) the licence fee paid to the firm by each user. When n^P ($n^P \in [0,1]$) figures the proportion of users of proprietary software and (q, c_L) the quality and the level of fees chosen by the firm, the profit maximization program of the firm is defined by (1):

$$\max_{c_L \geq 0, q \in [0,1]} \pi = \sup (0, n^P c_L - \beta q^2) \quad (1)$$

2.4. Utility generated by proprietary software adoption

The ability of both proprietary software and open source software to perform generic functionalities depends strongly on the total number of adopters. This network externality is motivated by several factors. For instance, compatibility among users and availability of complementary services and software are two examples of cases that are likely to lead to the development of a positive network externality. We suppose, therefore, that the utility associated with these functionalities depends on the number of users of the software solution considered. Consequently, we assume that the utility generated by the general needs $f_1(n^P)$ positively depends on the level of adoption n^P of proprietary software (with $f_1'(\cdot) > 0$, $f_1(0) = 0$ and $f_1(1) = \alpha_1$, $\alpha_1 > 0$). For convenience, we set $f_1(n^P) = \alpha_1 n^P$.

Users are heterogeneous regarding specific needs. Let us assume that the value they derive from specific functionalities is normalized to d ($d > 0$) for any type of software. Since software is mainly an experience good, risk-neutral users need to anticipate the value they derive from specific functionalities inasmuch as they cannot perfectly anticipate the whole set of functionalities they are likely to use once the software solution is adopted. In the case of proprietary software, the editor develops a proportion q of specific functionalities which are known by all users, and users expect a level of utility defined so that $dq + 0(1 - q) = dq$ (cf. figure 1). Since the source code is closed, the range of the functionalities available is strictly ruled by the firm.

Once the licence cost c_L charged by the firm has been deduced, the net utility derived from the adoption of proprietary software can be expressed as follows:

$$u_i^P = \alpha_1 n^P + dq - c_L \quad (2)$$

2.5. The utility generated by open source software adoption

The adoption of open source software does not involve a licence fee. As already assumed in the case of proprietary software adoption, we suppose that open source software meets all the generic functionalities required for adop-

FIGURE 1 : The utility of proprietary software

tion (3). As already mentioned, generic functionalities are subject to a direct externality. Also, developers – in the case of open source software – can freely modify and customize the initial software solution. The modular architecture of most contemporary open source software invites customization allowing open source developers to add functionalities that will benefit all users. Since open source software is maintained and managed by developers, the utility derived from specialized functionalities is highly dependent on the number of contributors (n_D^{OS}). In other words, the larger the number of developers who contribute to the code, the more functionalities will be added and the higher will be the utility of the open source software for all users (*i.e.*, developers and end-users). We specify this second externality by $f_2(n_{OS}^D) = \alpha_2 \left(\frac{n_{OS}^D}{\mu} \right) d + \alpha_2 \left(1 - \frac{n_{OS}^D}{\mu} \right) (0) = \alpha_2 \left(\frac{n_{OS}^D}{\mu} \right) d$. Once again, users gain a benefit d from specific functionalities only if these functionalities fulfil an expressed need. For risk-neutral users, this occurs at a probability of level $\alpha_2 \left(\frac{n_{OS}^D}{\mu} \right)$, $n_{OS}^D \in [0; \mu]$. This

- (3) We may suppose that these functionalities are *ex ante* developed by « kernel » developers and/or by the project manager so that open source software exists once all these functionalities are fully available (*e.g.*, « 1.0 » software version release).

probability is similar to the index of quality for proprietary software. However, the number of functionalities depends here on the number of developers within the open source community contribute to the code. The number of developers is set to be, at most, equal to μ , leading to a relative share of – active – developers equal to $(n_{OS}^D/\mu) = 1$. α_2 ($\alpha_2 \in]0;1]$) is the quality of the managerial model of the open source community. If $\alpha_2 = 1$ and all the developers act within the community, all the functionalities are developed (i.e., $\alpha_2 \left(\frac{n_{OS}^D}{\mu} \right) d = d$). Else (4), $\alpha_2 < 1$ and a smaller share of all the potential functionalities is eventually developed. It should be noted that μ is an exogenous parameter which depicts the share of developers within the whole community. Hence, the open source software project cannot rule that share. Moreover, our specification of the development externality captures the fact that the share of the functionalities developed depends on the proportion of developers actively involved in the open source project (i.e., n_{OS}^D/μ).

The adoption cost c_i^{EU} ($0 \leq c_i^{EU} \leq \bar{c}^{EU}$) which is supported by any user i ($i \in [0,1]$), derives from the user's gross utility. Developers and end-users are classified according to the level of their adoption costs. Consequently, user i is a developer and his adoption cost is null when $i \in [0, \mu]$. End-users are ranked

FIGURE 2: Open source software-related adoption costs

- (4) E.g., there might be persistence of coordination costs within the community and provision of a less adequate leadership strategy.

FIGURE 3: Structure of the sequential game
(OS = Open Source Software adoption;
P = Proprietary Software adoption; ∅ = No adoption)

on the segment $i \in [\mu, 1]$) according their adoption costs which increase at a linear rate. By definition, the adoption cost of end-user i is $c_i^{EU} = \left(\frac{i-\mu}{1-\mu} \right) \bar{c}^{EU}$ (cf. figure 2).

We can express the utility associated with open source adoption by agent i ($i \in [0, 1]$) as follows:

$$u_i^{OS} = \alpha_1 n^{OS} + \alpha_2 \left(\frac{n_D^{OS}}{\mu} \right) d - c_i^A \quad \text{with} \quad c_i^A = \begin{cases} 0 & \text{when } i \in [0, \mu] \\ c_i^{EU} & \text{when } i \in [\mu, 1] \end{cases} \quad (3)$$

III. — THE SEQUENTIAL GAME

The firm knows the different characteristics of the potential users (μ , α_1 , $\alpha_2, \dots$) of the proprietary software. It sets out its strategy and users react to this strategy by formulating their adoption choices. The firm must anticipate the responses of developers and end-users to its price and quality decisions. It has complete, but imperfect information about the possible strategies of potential adopters of its proprietary software. Put differently, the firm knows the different characteristics of the potential users (μ , α_1 , $\alpha_2, \dots$) of the software solution, but it can only imperfectly anticipate their choices when there are many possible equivalent choices. Since we may suppose that the price and the quality of software cannot be altered at will, we assume that the firm's and the users' choices are sequential. The structure of the sequential game is depicted in figure 3.

At the first step of the game, the firm chooses the extent of functionalities q^* and the level of its licence fee c_L^* which maximize its expected profit. At the second step of the game, taking these levels into account, each user i compares its respective utilities as a proprietary software user (*i.e.*, « P » strategy), an open source software adopter (*i.e.*, « OS » strategy) or a non adopter (*i.e.*, « $\emptyset$ » strategy). The individual utilities of users depend on each user's expected level of adoption of each software proprietary or OS software. As a simplifying assumption, we suppose that these expectations – whether right or wrong – are the same for all users. An educative process generates convergence of wrong expectations towards the locally stable equilibrium in the area of stability to which they belong. The game is solved by backward induction.

3.1. Second step of the game

Let us thus consider couple (q, c_L) . Once we note the expectations $\tilde{n}^P, \tilde{n}^{OS}, \tilde{n}_D^{OS}$ relative to the respective sizes of the population of proprietary software customers, the population of open source software adopters and the population of developers who contribute code within open source projects, the choice of any user i ($i \in [0,1]$) is as follows :

- choosing P if $\alpha_1 \tilde{n}^P + dq - c_L \geq \sup \left\{ 0, \alpha_1 \tilde{n}^{OS} + \alpha_2 \left(\frac{\tilde{n}_D^{OS}}{\mu} \right) d - c_i^A \right\}$
- choosing OS if $\alpha_1 \tilde{n}^{OS} + \alpha_2 \left(\frac{\tilde{n}_D^{OS}}{\mu} \right) d - c_i^A \geq \sup \left\{ 0, \alpha_1 \tilde{n}^P + dq - c_L \right\}$
- choosing $\emptyset$ if $0 \geq \sup \left\{ \alpha_1 \tilde{n}^P + dq - c_L, \alpha_1 \tilde{n}^{OS} + \alpha_2 \left(\frac{\tilde{n}_D^{OS}}{\mu} \right) d - c_i^A \right\}$

Each agent associates its own optimal choices to each expected level of $\tilde{n}^P, \tilde{n}^{OS}$ and $\tilde{n}_D^{OS}$. Let us assume conventionally that, if agents expect the level $\tilde{n}^{OS}$, they will consistently suppose that this population only includes developers if $\tilde{n}^{OS} \leq \mu$. If $\tilde{n}^{OS} > \mu$, this population also integrates the end-users whose adoption costs are the lowest (5). Aggregation of these optimal decisions allows us to deduce the effective proportions of proprietary software users, as well as both the developers and end-users of open source software (*i.e.*, n^P, n^{OS} and n_D^{OS} respectively), by the expected levels of each populations (see Appendix 1). Hence, for each pair (q, c_L) , the condition $(n^P, n^{OS}, n_D^{OS}) = (\tilde{n}^P, \tilde{n}^{OS}, \tilde{n}_D^{OS})$ defines the Nash equilibrium of the subgame corresponding to the second step of the game.

(5) This kind of assumption is frequently used in adoption models (see *e.g.*, Crémer, 2000).

Proposition 1. *Whatever the pair (q, c_L) that is selected by the firm at the first step of the game, there exists at least one Nash equilibrium associated with the second-step sub-game.*

Proof of Proposition 1. See Appendix 1.

Without any specific restriction on the values of the parameters, the competition between open source and proprietary software could provide some tautological and irrelevant outcomes. Since we are first interested in competitive issues, let us consider the following non restrictive assumption.

Assumption 1. *The activity of the proprietary software firm is profitable when the open source development activity is not introduced.*

Assumption 1 only states that in the absence of an open source competitor, the commercial player can always get a positive market share and earn some positive profit. Put differently, Assumption 1 states that there is an economically viable pair (q^+, c_L^+) , i.e., a quality/price combination for the proprietary software firm, so that i) there is an expected size level $\tilde{n}^{P+}$ of proprietary software users which is high enough to provide positive incentives for users to adopt proprietary software when there is no alternative, and ii) for this pair (q^+, c_L^+) , the profit of the firm $\pi = c_L^+ - \beta (q^+)^2$ is non-negative. Proposition 2 derives from Assumption 1.

Proposition 2. *Under Assumption 1, the distribution of users $\{n^{P*} = 1, n_D^{OS*} = 0, n^{OS*} = 0\}$ which corresponds to the full adoption of proprietary software is always a Nash equilibrium of the second-step subgame.*

Proof of Proposition 2. See Appendix 1.

Proposition 2 establishes that the viability of proprietary software alone is a sufficient condition to generate a Nash equilibrium in which open source software is always crowded out. In that case, users do not adopt the open source software.

With Assumption 1, the second-step subgame has at least one equilibrium. At this equilibrium, the open source development activity is crowded out and all the users adopt proprietary software. But there may be other equilibria. Consider for instance the extreme case in which adoption cost c_i^A vanishes whatever i is. With a sufficiently low value of q and a sufficiently high value of c_L , inequality $f_1(0) + dq - c_L \leq 0 \leq f_1(1) + dq - c_L$ may hold. Since $f_1(1) + f_2(\mu) - c_i^A$ is non-negative in this case whatever i , condition $\sup \{0, f_1(0) + dq - c_L\} \leq f_1(1) + f_2(\mu) - c_i^A$ also applies to all the users. Outcome $\{n^{P**} = 0, n_D^{OS**} = \mu, n^{OS**} = 1\}$ is thus a second equilibrium in the second-step subgame. By continuity, the same result is maintained when $c_i^A = \varepsilon_i (\forall i)$ with ε_i close to 0. Hence, $\{n^{P**} = 0, n_D^{OS**} = \mu, n^{OS**} = 1\}$ exists for a non-empty range of variation of $c_i^A, \forall i$. However, as c_i^A is far from being null for a part of the popula-

tion of end-users, this « Winner-Takes-All » (now, WTA) – open source – equilibrium does not exist for all possible values of $\bar{c}^{EU}$. Last, for some values of $\bar{c}^{EU}$, other equilibria may exist. As opposed to WTA equilibria $\{n^{P*} = 1, n_D^{OS*} = 0, n^{OS*} = 0\}$ and $\{n^{P**} = 0, n_D^{OS**} = \mu, n^{OS**} = 1\}$, these equilibria are « mixed », that is to say, they split users into two sub-populations, namely users of proprietary software and users of open source software (6).

We then introduce a second reasonable assumption parallel to Assumption 1.

Assumption 2. *The higher level of adoption costs that users may face is upper bounded so that $\bar{c}^{EU} \leq (\alpha_1 + d\alpha_2)$.*

This assumption supposes that the end-user who exhibits the largest adoption cost $\bar{c}^{EU}$ always gets a non-negative utility from adopting open source software. This assumption is similar to the conditions that a proprietary software firm faces for its activity to remain profitable.

We add a third assumption for the expectations of the agents.

Assumption 3. *Agent i cannot simultaneously adopt open source software and expect that a share of agents less than i adopt it.*

Assumption 3 excludes some cases in which expectations and choices will not be consistent. This leads us to suppose that agents are aware of their ranking among other users and are likely to integrate this information in their adoption decisions by excluding expectations that are inconsistent with their rankings.

Under these three assumptions, we can analyse in depth both the nature and the stability of the second-step subgame equilibria (see Appendix 1). At this stage, the firm has already set c_L and q . The remaining variables which have to be determined by the users are n^{P*} (i.e., the Nash equilibrium proportion of agents adopting proprietary software), n^{OS*} (i.e., the Nash equilibrium proportion of agents adopting open source software) and n_D^{OS*} (i.e., the Nash equilibrium proportion of developers among agents adopting open source software). According to our specifications of $f_1(\cdot)$, $f_2(\cdot)$ and $g(\cdot)$, and given Assumptions 1, 2 and 3, the typology of the Nash equilibrium in this second-step subgame can be analysed by considering all the possible values of parameters α_1 , α_2 , $\bar{c}^{EU}$, μ and d , as well as the first-step control variables (i.e., q and c_L). Table 1 summarizes the three main cases we identify. Appendix 1 presents the seven subcases we obtain from our analysis and their graphical representations.

- (6) Note that there may be many separating equilibria in a more general case when functions $f_1(\cdot)$ and $f_2(\cdot)$ are not linear and function $g(\cdot)$ is not quadratic. Although our linear-quadratic specification limits the range of possibilities, the multiplicity of Nash separating equilibria is a possible outcome in the second-step subgame.

TABLE 1 : Nash equilibria of the second step of the game

	Adoption patterns	Number of equilibria
Case A (subcases 1, 4, 2 and 5)	WTA – Proprietary $\{n^P* = 1, n_D^{OS*} = 0, n^{OS*} = 0\}$	3 equilibria
	WTA – Open source $\{n^P* = 0, n_D^{OS*} = \mu, n^{OS*} = 1\}$	2 stable equilibria
Case B (subcases 3 and 6)	WTA – Proprietary $\{n^P* = 1, n_D^{OS*} = 0, n^{OS*} = 0\}$	1 stable equilibria
Case C (subcase 7)	WTA – Proprietary $\{n^P* = 1, n_D^{OS*} = 0, n^{OS*} = 0\}$	3 equilibria
	Mixed – Proprietary and Open source $\{n^P* < 1, n_D^{OS*} = \mu, n^{OS*} < 1\}$	2 stable equilibria

The index of each subcase refers to Appendix 1

In case A, there are three Nash equilibria, only two of which are stable. In this case, either proprietary software or open source software is adopted at the equilibrium, but both types of software never simultaneously co-exist at the end of educative process. These cases exhibit a WTA situation. The area of attraction of this type of equilibrium is defined by the location of the unstable « mixed » equilibrium (E3, see Appendix 1). When the initial users' expectations about the adoption of open source software are below a critical level, proprietary software is eventually adopted by all the users at the end of the educative process. In contrast, open source software will be adopted if expectations are beyond this critical level.

Case B corresponds to a situation in which there is a single stable WTA equilibrium situation in which either proprietary software or open source soft-

ware is adopted. Finally, Case C corresponds to a situation in which there are two stable equilibria. As already emphasized, the first equilibrium is still a WTA equilibrium according to which all users adopt proprietary software. However, the second equilibrium depicts patterns of mixed adoption because some users adopt open source software and others adopt proprietary software. All things being equal, case C occurs for low levels of adoption costs and high levels of quality related to the managerial model of the open source community (*i.e.*, α_2).

These results stand for one given (q, c_L) strategy of the firm. Solving backward, we define the equilibrium in the first-step subgame that maximizes the profit of the proprietary software firm.

3.2. First step of the game

The first step of the game corresponds to the choice of the pair (q, c_L) which maximizes the expected profit of the firm, given the *ex post* adoption scenarios that are most likely to apply.

At this stage, the firm has to choose the pair (q, c_L) which maximizes its profit function. As the second-step subgame equilibrium is generally not unique, the firm has to integrate each possible stable outcome which is associated with the second step of the game that results from pair (q, c_L) . The occurrence of these stable equilibria and their related payoffs also have to be considered. The behaviour of the firm towards risk may influence the shape of its objective function. For simplicity, we suppose that the firm is risk-neutral and only considers the expected profit that is generated by the different outcomes of the second-step subgame as answers to its choices at the first step of the game (7). The expected profit of the firm associated with a given pair (q, c_L) uses an objective or subjective distribution of probability on the occurrence of the stable equilibria of the second-step subgame. The form of this distribution obviously influences the firm's decision. As the firm's information is imperfect, it reasonably anticipates a uniform distribution $[0,1]$ of the level of open source software adoption $\tilde{n}^{OS}$. We label $\{n^{P^+}, n_D^{OS^+}, n^{OS^+}\}$ (resp. $\{n^{P^{++}}, n_D^{OS^{++}}, n^{OS^{++}}\}$) the population that corresponds to an unstable (resp. stable) second-step Nash equilibrium. From Assumption 3, the size of the respective areas of stability of the second-step subgame equilibria provides the required distribution of probability which generates the expected profit of the firm, which is expressed by (4):

$$\pi = \max_{c_L \geq 0, q \in [0,1]} [0, (p^* + p^{++} n^{P^{++}}(c_L, q)) c_L - \beta q^2] \quad (4)$$

- (7) We make this assumption because the probability of an unstable equilibrium occurring is zero in this context.

Where :

- $p^* = 1$ and $p^{\dagger\dagger} = 0$ when (q, c_L) and $(\alpha_1, \alpha_2, d, \mu, \bar{c}^{EU})$ correspond to subcases (3) and (6) (see Appendix 1),
- $p^* = n^{P\dagger}$ and $p^{\dagger\dagger} = (1 - n^{P\dagger})$ when (q, c_L) and $(\alpha_1, \alpha_2, d, \mu, \bar{c}^{EU})$ correspond to subcases (1), (2), (4) and (5) (i.e., $n^{P\dagger\dagger}(c_L, q) = 0$, see Appendix 1),
- $p^* = n^{P\dagger}$ and $p^{\dagger\dagger} = (1 - n^{P\dagger})$ when (q, c_L) and $(\alpha_1, \alpha_2, d, \mu, \bar{c}^{EU})$ correspond to subcase (7) (i.e., $0 < n^{P\dagger\dagger}(c_L, q) < 1$, see Appendix 1).

We can deduce the existence of an optimal outcome for the firm at the first step of the game and that of an equilibrium in the two-step game.

Proposition 3. *An optimal outcome always exists for the firm at the first step of the game.*

Proof of Proposition 3. See Appendix 2.

Corollary. An equilibrium of the two-step game always exists. When this equilibrium is not unique, equilibria correspond – except for cases of vanishing measure – to a single level of both functionalities and licence cost of proprietary software, but leads to different levels of adoption of proprietary software and open source software.

Comparison among subcases 1, 2 and 3 illustrates the nature of the choice made by the firm at the first step of the game. In these three cases, the same condition $\alpha_1(1 - \mu) > \bar{c}^{EU}$ is met on three out of five parameters. According to the last two, d and α_2 , which characterize the efficiency of the effort of the firm and that of adoption externalities, the firm decides to choose a rather low level of quality and a high level of price with a probability of success that is less than 1 in subcases 1 and 2 or a rather high level of quality and low level of price with a probability of success that is equal to 1 in subcase 3.

In subcases 1 and 2, the level of profit of the firm may be large if success eventually applies, but this success may not occur. In subcase 3, profit is lower, but the firm can always increase this level.

We next present three numerical examples to analyse possible competition outcomes.

3.3. Some illustrative examples

The first example considers values of parameters corresponding to case A ($\alpha_1 = 1, d = 1, \alpha_2 = 1, \beta = 0.2, \mu = 0.2, \bar{c}^{EU} = 0.1$). In this case, direct externality $f_1(.)$ outweighs the adoption cost incurred by the marginal user. Remember, that the final outcome is always a winner-takes-all (WTA) situa-

FIGURE 5: Expected profit of the firm (left) and expected diffusion of open source software (right) as a function of the quality and the license cost charged by the firm
 $(\alpha_1 = 1, d = 1, \alpha_2 = 1, \beta = 0.2, \mu = 0.2, \bar{c}^{EU} = 0.1)$;
 $\pi^{\max} = 0.043 > 0$ reached for $(c_L^* = 0.6$ and $q^* = 0.2)$

tion, which leads to the crowding out of one of the two software producers. We plot the firm's expected profit for different (q, c_L) strategies ; figure 5 presents the results (8).

The maximum expected profit (denoted $\pi^{\max}$) here is strictly positive and reaches an interior value of the strategy set ($c_L^* = 0.6$ and $q^* = 0.2$). The right hand side figure depicts the expected diffusion of open source software, *i.e.*, the probability that open source software is diffused to the whole population. It shows how the proprietary firm can indirectly – and partially – control the

- (8) For each strategy, we consider the relative positions of the U^P and U^{OS} curves. From this comparison, we deduce the most suitable subcase (*i.e.*, subcase 1, 2 or 3) and we can compute the expected profit and the expected diffusion of the open source project. If $U^P < 0$ ($\forall i \in [0,1]$), no user adopts proprietary software. Therefore, the firm does not make any investment on quality-based features and the profit of the firm is null.

diffusion of open source software by setting different price and quality strategies. When the firm chooses a « low price-high quality » strategy, it can crowd the open source community out of the market (*i.e.*, the expected diffusion of open source software is close to 0). However, this example illustrates that the firm has no incentive to do this due to the development costs. Instead, the expected diffusion of open source software which corresponds to the optimal strategy of the firm is found to be equal to 0.91. Here, the firm develops only 20 % of the full set of functionalities it could develop.

Thus, the firm faces a trade-off inasmuch as it may select a strategy that crowds out the open source project by setting a high quality and/or low license cost, or it may apply a « low quality and/or high price » strategy. In the latter case, the firm gets a maximal profit when the WTA – proprietary – equilibrium is reached, but it risks being crowded out (*i.e.*, open source software is massively adopted). The optimal behaviour of the firm has to be understood as a balance between these two strategies. We could mitigate this result by taking account of the firm's risk aversion. However, we suggest that this change would not change our results qualitatively. Indeed, a more risk-averse firm would be likely to minimize the risk of being crowded by applying an optimal « conservative » strategy (*i.e.*, a quality increase and/or price decrease strategy).

The second example depicts a situation in which the presence of specialized functionalities is very important for shaping adoption patterns (see figure 6). In this example, we set $d = 5$ instead of $d = 1$.

The profit-maximizing strategy here aims at developing a proprietary software solution which provides the largest range of specialized functionalities ($q^* = 1$). Hence, the firm has an incentive to invest in quality in order to diffe-

FIGURE 6 : Expected profit of the firm (left) and expected diffusion of open source software (right) as a function of the quality and the license cost charged by the firm
 $(\alpha_1 = 1, d = 5, \alpha_2 = 1, \beta = 0.2, \mu = 0.2, \bar{c}^{EU} = 0.1)$;
 $\pi^{\max} = 0.08 > 0$ reached for $(c_L^* = 2.5$ and $q^* = 1)$

rentiate its software from open source software. We note that the expected diffusion of open source software is equal to 0.88. As such, the optimal strategy of the firm is not to crowd the open source community out of the market.

The third example presents a situation in which users' needs are rather generic (see figure 7). Here, the presence of general functionalities shapes adoption outcomes. Preferences for specific needs are lower-level so that we set $d = 0.1$.

FIGURE 7: Expected profit of the firm (left) and expected diffusion of open source software (right)

as a function of the quality and the license cost charged by the firm

$$(\alpha_1 = 1, d = 0.1, \alpha_2 = 1, \beta = 0.2, \mu = 0.2, \bar{c}^{EU} = 0.1) ;$$

$$\pi^{\max} = 0.124 > 0 \text{ reached for } (c_L^* = 0.4 \text{ and } q^* = 1)$$

Our results show that the firm does not develop any specific functionality ($q = 0$). Since users assign little importance to specific functionalities, the firm's optimal strategy is to focus on generic functionalities and not to invest in specialized ones. As the first two examples showed, we find that the firm has no interest in crowding out the open source activity.

IV. — MARKET FAILURE AND THE WELFARE PROPERTIES OF THE INTRODUCTION OF OSS

The findings from the previous analysis show situations in which multiple equilibria are likely to prevail as a result of both network and compatibility externalities. The nature of the equilibrium that emerges, therefore, is related to the shape of users' expectations.

On the basis of the previous results, we perform two types of welfare analysis. We need first to deal with the multiplicity of equilibria identified. We have shown that a single firm strategy is likely to lead to the appearance of multiple equilibria. So is it possible to rank potential outcomes? Second, we can track the aims of these outcomes to those likely if the firm acts as a monopolist, that

is to say, if there are no open source communities. We conduct these welfare analyses on our linear-quadratic specification.

When the optimal pair (q^*, c_L^*) leads to the occurrence of multiple equilibria, market failure may emerge. For instance, in the linear-quadratic specification, consider the case where (q^*, c_L^*) generates two stable WTA – proprietary software and open source software – equilibria, *i.e.*, $\{n^{p*} = 1, n_D^{OS*} = 0, n^{OS*} = 0\}$ and $\{n^{p**} = 0, n_D^{OS**} = \mu, n^{OS**} = 1\}$. Comparison of these WTA equilibria provides the following result in the linear-quadratic case.

Proposition 4. *When the optimal price/quality pair (q^*, c_L^*) is associated with two stable WTA equilibria at the second-step subgame,*

- i) if a coordination failure occurs, the higher-level equilibrium is always the WTA – proprietary – one. All things being equal, this case prevails when development externalities are low-level;*
- ii) if a conflict of interest occurs, this conflict rises between developers and part of or all the end-users on one hand and the firm with – or without – some other end-users on the other hand. All things being equal, this case prevails when development externalities are high-level.*

Proof of Proposition 4. See Appendix 3.

The first part of Proposition 4 indicates that there are cases where a low level of expectations related to the adoption of proprietary software may eventually generate a WTA – open source – solution that is not beneficial even to developers. In these cases, all users join the open source community due to the shaping of users' expectations. However, as the quality of the managerial model of the project (*i.e.*, α_2) is low, a low share of functionalities is developed. The second part of Proposition 4 establishes that the WTA – open source – solution (resp. WTA – proprietary – solution) delivers the better outcome for both developers and some end-users (resp. the firm and the other end-users).

These results may be extended by a complementary proposition where one of the stable solutions is a mixed equilibrium.

Proposition 5. *When the optimal price/quality pair (q^*, c_L^*) is associated with one WTA – proprietary – solution and one mixed equilibrium at the second-step sub-game,*

- i) if a coordination failure occurs, the higher-level equilibrium is always the WTA – proprietary – one. All things being equal, this case prevails when development externalities are low-level;*
- ii) if a conflict of interest occurs, this conflict rises between developers and a part of all the end-users on the one hand, and the firm and the other end-users on the other hand. All things being equal, this case prevails when development externalities are high-level.*

Proof of Proposition 5. See Appendix 3.

Proposition 5 confirms Proposition 4. Due to the existence of the externalities generated by each type of software, when the equilibria can be ranked according to welfare, the mixed equilibrium is not efficient even if the firm is not considered in the welfare analysis. Propositions 4 and 5 show that an even larger diffusion of open source software may not be beneficial to users. However, these two propositions enable us to identify some cases in which this diffusion may be detrimental to the users' surpluses. Furthermore, Proposition 5 stresses that the incentives of some users and the firm may be aligned in some situations. Propositions 4 and 5 are in contrast to the following.

Proposition 6. *Suppose that a (unique or not) WTA – proprietary – equilibrium is associated with the optimal price/quality pair (q^*, c_L^*) in a framework in which adopters are able to choose between proprietary and open source software. All users prefer this equilibrium to the WTA – proprietary – one that emerges in a framework in which they cannot choose open source software.*

Proof of Proposition 6. See Appendix 3.

Proposition 6 counterbalances Propositions 4 and 5. Even if open source software is not eventually selected or adopted, the existence of a « potential competitor » for proprietary software provides strong incentives for the producer to lower the price or to improve the quality of its product. The ensuing outcome is always found to be beneficial to all the users. Propositions 4 and 5 compare the properties of the two possible equilibria when the open source community already acts in the software market. They show that the magnitude of the development externality is critical to identify the agents that benefit from the equilibrium eventually selected. Moreover, when there is a conflict of interest, we show that it does not affect the firm and all users, but rather that the interests of some users are aligned to those of the firm. In particular, we find that users whose adoption costs are high have the same interest than the editor, whereas the users whose adoption are low are aligned to the interests of the open source community.

V. — DISCUSSION AND FURTHER RESEARCH

This paper analysed the competition dynamics and coordination issues likely to arise from the introduction of open source solutions in the software market. We aimed to identify how a proprietary software firm defines its price/quality strategy by taking account of lock-in effects and users' expectations as well as the varying abilities of adopters to contribute or not to code in open source projects. Our results show that multiple – winner-takes-all and mixed – equilibria may emerge from the competition game between firms and the open source community. Moreover, depending on the users requirements for software functionalities, we show there are cases where i) the firm has no interest

in investing in quality or ii) has incentives to develop all the functionalities to maximize its profit.

A striking result is that the firm may have a rational (*i.e.*, market-based) interest in not trying to crowd the open source community out of the market (accommodating strategy), inasmuch as it is limited in its abilities to fully select the equilibrium which allows it to reach out the highest-level outcomes. Also, the welfare analysis shows that the open source « threat » is likely to increase levels of social surplus because it provides strong incentives for the firm to lower its price or to improve the quality of its product. As a consequence, the software market may be seen as an hybrid « system » in which traditional commercial firms have to adapt to the development of atypical production activities. Whatever the final outcome, be it market-shared or not, our findings suggest how commercial players need to revise their strategies for their presence to be sustainable in the long run. Such results relates to previous results obtained in the literature on competition between open source and proprietary software and also focuses on new insights (entry threat and welfare analysis) less analyzed in previous papers.

These findings indicate first that we need further numerical exploration of different scenarios. One direction is related to how the efficiency of the organization of open source projects – captured by α_2 in our model – impacts on their potential diffusion. Within our framework, this impact could be analysed by studying the reaction of the firm to alternative degrees of efficiency. Second, we could extend the analysis by considering an additional step in which the firm revises its price-quality strategy. This could be done as a repeated game. However, this would raise additional problems since many choices are not easily reversible (*e.g.*, adoption of users and quality of the firm).

Our focus in this paper was clearly on competitive linkages between open source and proprietary software. However, we might also consider alternative strategies related to the commercial firm's quality policy. For example, Lerner and Tirole (2001), Dahlander (2007) and Rolandsson *et al.* (2011) argue that firms can understand competition better by allowing their own developers to be involved in open source projects, by innovating as well as by using the open source ideology to improve their production levels and to detect skilled developers. Dahlander and Magnusson (2005; 2008) and Stam (2009) stress that firms may benefit directly from open source communities by managing their activities because the knowledge required to develop software will then be controlled by both the firm and the surrounding communities. As the addition of a specific functionality may be too costly, Krishnamurthy (2003), Bonaccorsi *et al.* (2006) and Dahlander and Wallin (2006) point out that firms should be encouraged to support open source projects to lower their production costs. In our framework, the open source « threat » may also provide the proprietary firm with assets (*e.g.*, source code, developers, toolkits) allowing it to improve the quality of its product at lower cost. Proprietary firms may then be able to benefit from the increasing popularity of some open source projects

and to develop new approaches to generate value from atypical organizations by switching to a « hybrid business model ». Departing from a pure competition viewpoint, such arguments leave room for further analysis of new forms of cooperation between open source communities and commercial players.

REFERENCES

- BESSEN J. (2002), « What good is free software ? », in Hahn R.-W. (Ed.), *Government policy toward open source software*, Brookings Institution Press, Washington, DC, pp. 12-33.
- BITZER J. (2004), « Commercial versus open source software: The role of product heterogeneity in competition », *Economic Systems* 28(4), 369-381.
- BONACCORSI A., GIANNANGELI S., ROSSI C. (2006), « Entry strategies under competing standards: Hybrid business models in the open source software industry », *Management Science* 52(7), 1085-1098.
- BONACCORSI A., ROSSI C. (2003), « Why open source software can succeed », *Research Policy* 32(7), 1243-1258.
- CASADESUS-MASANELL R., GHEMAWAT P. (2006), « Dynamic mixed monopoly: A model motivated by Linux vs. Windows », *Management Science* 52(7), 1072-1084.
- CRÉMER J. (2000), « Network externalities and universal service obligation in the internet », *European Economic Review* 44(4-6), 1021-1031.
- DAHLANDER L. (2007), « Penguin in a new suit: A tale of how de novo entrants emerged to harness free and open source software communities », *Industrial and Corporate Change* 16(5), 913-943.
- DAHLANDER J., MAGNUSSON M.-G. (2005), « Relationships between open source software companies and communities: Observations from nordic firms », *Research Policy* 34(4), 481-493.
- DAHLANDER L., MAGNUSSON M. (2008), « How do firms make use of open source communities ? », *Long Range Planning* 41(6), 629-649.
- DAHLANDER L., WALLIN M.-W. (2006), « A man on the inside: Unlocking communities as complementary assets », *Research Policy* 35(8), 1243-1259.
- DALLE J.-M., JULLIEN N. (2000), « NT vs. Linux, or some explorations into the economics of free software », in Ballot G., Weisbuch G., (Eds.), *Application of Simulation to Social Sciences*, Hermès, Paris, France, pp. 399-416.
- DALLE J.-M., JULLIEN N. (2003), « “Libre” software: Turning fads into institutions ? », *Research Policy* 32(1), 1-11.
- ECONOMIDES N., KATSAMAKAS E. (2006), « Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry », *Management Science* 52(7), 1057-1071.
- FRANKE N., von HIPPEL E. (2003), « Satisfying heterogeneous user needs via innovation toolkits: The case of Apache security software », *Research Policy* 32(7), 1199-1215.
- GHOSH R.-A., HAALAND K., HALL B.-H. (2008), *Which firms participate in open source development ? A study using data from Debian*, DIME Working Papers on Intellectual Property Rights 73.
- HARHOFF D., MAYRHOFER P. (2008), *User communities and hybrid innovation processes. Theoretical foundations and implications for policy and research*, DIME Working Papers on Intellectual Property Rights 85.
- HENKEL J. (2004), *The jukebox mode of innovation – A model of commercial open source development*, CEPR Discussion Papers 4507.

- HERTEL G., NIEDNER S., HERRMANN S. (2003), « Motivation of software developers in open source projects: An internet-based survey of contributions to the Linux kernel », *Research Policy* 32(7), 1159-1177.
- JOHNSON J.-P. (2002), « Open source software: Private provision of a public good », *Journal of Economy and Management Strategy* 11(4), 637-662.
- KRISHNAMURTHY S. (2003), « A managerial overview of open source software », *Business Horizons* 9-10, 47-56.
- KUAN J. (2002), *Open source software as lead user's make or buy decision: A study of open and closed source quality*, Working paper, Stanford University, Stanford, CA.
- LANZI D. (2009), « Competition and open source with perfect software compatibility », *Information Economics & Policy*, 21(3), 192-200.
- LEONCINI R., RENTOCCHINI F., VITTUCCI MARZETTI G. (2008), *You won the battle. What about the war? A diffusion model on proprietary-open source software competition*, Working paper, University of Bologna, Bologna, Italy.
- LERNER J., TIROLE J. (2001), « The open source movement: Key research questions », *European Economic Review* 45(4-6), 819-826.
- LERNER J., TIROLE J. (2002), « Some simple economics of open source », *Journal of Industrial Economics* 50(2), 197-234.
- MUSTONEN M. (2003), « Copyleft – The economics of Linux and other open source software », *Information Economics and Policy* 15(1), 99-121.
- RIEHLE D. (2009), « The commercial open source business model ». *Proceedings of the 15th Americas Conference on Information Systems*, AIS Electronic Library, Paper 104.
- ROLANDSSON B., BERGQUIST M., LJUNGBERG J. (2011), « Open source in the firm: Opening up professional practices of software development », *Research Policy* 40(4), 576-587.
- SCHMIDT K., SCHNITZER M. (2003), « Public subsidies for open source? Some economic policy issues of the software market », *Harvard Journal of Law and Technology* 16(2), 473-505.
- STAM W. (2009), « When does communities participation enhance the performance of open source software companies? », *Research Policy* 38(8), 1288-1299.
- VÄLIMÄKI M., OKSANEN V. (2005), « The impact of free and open source licensing on operating system software markets », *Telematics and Informatics* 22(1-2), 97-110.
- von ENGELHARDT S., MAURER S.-M. (2010), *The new (commercial) open source: Does it really improve welfare?*, Paper presented at the 4th FLOSS International Workshop on Free/Libre Open Source Software, Jena, Germany.
- von KROGH G., von HIPPEL E. (2006), « The promise of research on open source software », *Management Science* 52(7), 975-983.
- WEST J., LAKHANI K.-R. (2008), « Getting Clear About Communities in Open Innovation », *Industry & Innovation* 15(2), 145-168

APPENDIX 1

NASH EQUILIBRIA OF THE SECOND-STEP SUBGAME

Proof of Proposition 1: Given that $\tilde{n}_D^{OS} = \inf [\tilde{n}^{OS}, \mu]$, let us consider the choice which is made by agent i when he observes pair (q, c_L) at the second step of the game:

- Choosing P if $\alpha_1 \tilde{n}^P + dq - c_L \geq \sup \left\{ 0, \alpha_1 \tilde{n}^{OS} + \alpha_2 \left(\frac{\tilde{n}_D^{OS}}{\mu} \right) d - c_i^A \right\}$

- Choosing OS if $\alpha_1 \tilde{n}^{OS} + \alpha_2 \left(\frac{\tilde{n}_D^{OS}}{\mu} \right) d - c_i^A \geq \sup \{ 0, \alpha_1 \tilde{n}^P + dq - c_L \}$
- Choosing $\emptyset$ if $0 \geq \sup \left\{ \alpha_1 \tilde{n}^P + dq - c_L, \alpha_1 \tilde{n}^{OS} + \alpha_2 \left(\frac{\tilde{n}_D^{OS}}{\mu} \right) d - c_i^A \right\}$

The analysis of the second-step subgame consists of proving the existence of fixed points to correspondence $\tau = \{t^P, t^{OS}, t^\emptyset\}$ from $\{[0,1], [0,1], [0,1]\}$ to itself, where τ associates – for all the values of parameters $(\alpha_1, \alpha_2, d, \mu, \bar{c}^{EU})$ and variables (q, c_L) – the actual population $(n^P, n^{OS}, n^\emptyset)$ of adopters with the set $(\tilde{n}^P, \tilde{n}^{OS}, \tilde{n}^\emptyset)$ which corresponds to the same expected populations. Since $\tilde{n}^\emptyset$ and $n^\emptyset$ are defined so that $\tilde{n}^\emptyset = 1 - \tilde{n}^P - \tilde{n}^{OS}$ and $n^\emptyset = 1 - n^P - n^{OS}$, correspondence τ has the same properties as correspondence $\tau' = \{t^P, t^{OS}\}$ from $\{[0,1], [0,1]\}$ to itself which associates the pair of effective populations (n^P, n^{OS}) to the pair of expected populations $(\tilde{n}^P, \tilde{n}^{OS})$.

As τ' is defined from a compact set to itself, the continuity of t^P and t^{OS} is a sufficient condition to prove the existence of at least one fixed point of τ' from which we can then deduce the existence of a fixed point of τ . Whatever $(\alpha_1, \alpha_2, d, \mu, \bar{c}^{EU})$, (q, c_L) and $(\tilde{n}^P, \tilde{n}^{OS})$ are, n^P corresponds to the subset of $i \in [0,1]$ which is defined so that $\alpha_1 \tilde{n}^P + dq - c_L \geq \sup \left\{ 0, \alpha_1 \tilde{n}^{OS} + \left(\frac{\alpha_2 d}{\mu} \right) \tilde{n}_D^{OS} - c_i^A \right\}$ and n^{OS} corresponds to the subset of $i \in [0,1]$ which is defined so that $\alpha_1 \tilde{n}^{OS} + \left(\frac{\alpha_2 d}{\mu} \right) \tilde{n}_D^{OS} - c_i^A \geq \sup \{ 0, \alpha_1 \tilde{n}^P + dq - c_L \}$. These subsets are closed ones in $[0,1]$ as well as their complementary $n^\emptyset$. Each term that defines n^P and n^{OS} is continuous on $\tilde{n}^P$ and $\tilde{n}^{OS}$, and n^P and n^{OS} are therefore continuous on $\tilde{n}^P$ and $\tilde{n}^{OS}$. Thus, $n^\emptyset$ is also continuous on $\tilde{n}^P$ and $\tilde{n}^{OS}$, and correspondences τ and τ' are shown to be continuous on $\tilde{n}^P$ and $\tilde{n}^{OS}$. Since these transformations are defined from a compact set to itself, they admit at least one fixed point. This (these) fixed point(s) is (are) the Nash equilibrium (equilibria) of the second-step subgame. Let us now suppose that $\{n^{P*}, n^{OS*}, n^{\emptyset*}\}$ is (one of) the fixed point(s) of this transformation. The size of population n_D^{OS*} can be then elementarily deduced from $n_D^{OS*} = \inf [n^{OS*}, \mu]$.

Proof of Proposition 2: Let us introduce (q^+, c_L^+) as an economically viable pair of quality and licence cost of proprietary software. When open source software is not available in the software market, since agents are not *ex ante* differentiated in their abilities to access to proprietary software, users $(1 - \tilde{n}^{P+})$ eventually adopt proprietary software if users $\tilde{n}^{P+}$ ($\tilde{n}^{P+}$ be a given value) find they would benefit from adopting proprietary software. Hence, if n^{P+} is a Nash equilibrium of the second-step subgame, $n^{P+} = 1$. Let us now consider that open source software applies in the software market for same pair (q^+, c_L^+) and let us suppose that the expected distribution of agents is given by $\{\tilde{n}^P, \tilde{n}_D^{OS}, \tilde{n}^{OS}\} = \{1, 0, 0\}$. The individual utilities which are generated by this distribution are $u^P = \alpha_1 + dq^+ - c_L^+ \geq \alpha_1 n^{P+} + dq^+ - c_L^+ \geq 0$ when user i adopts proprietary soft-

ware and $u_i^{OS} = -c_i^A \leq 0, \forall i$, when user i adopts open source software. For any user i , inequality $\alpha_1 \tilde{n}^P + dq - c_L \geq \sup \{0, \alpha_1 \tilde{n}^{OS} + \alpha_2 \left(\frac{\tilde{n}_D^{OS}}{\mu}\right) d - c_i^A\}$ holds and $\{n^P, n_D^{OS}, n^{OS}\} = \{1, 0, 0\}$. This proves that $\{n^{P*}, n_D^{OS*}, n^{OS*}\} = \{1, 0, 0\}$ is a Nash equilibrium for pair (q^+, c_L^+) in a context in which users can adopt open source software, which demonstrates Proposition 2.

Derivation of the Nash equilibria of the second-step game: The possible outcomes of the second-step subgame can be described according to the values of both parameters, $\alpha_1, \alpha_2, \bar{c}^{EU}, \mu$ and d , and variables q and c_L . For each set of values of these parameters and variables, we simultaneously use two related figures both of which capture the role of expectations on the individual utilities associated with the three adoption strategies (*i.e.*, proprietary software adoption, open source software adoption, and no-adoption). In the upper chart, we display the utility of the i^{th} agent respectively as a user of proprietary software and a user of open source software. Agents j ($j \in [0, i]$) adopt open source software and agents k ($k \in [i, 1]$) adopt proprietary software. As such, we designate agent i as « marginal » agent. Intersections here define interior solutions, that is to say, the equilibria that are defined so that one part of the agents adopts open source software whereas the other part of the agents adopts proprietary software. The situation according to which all the agents adopt proprietary software is always shown to be a solution when u_i^P – that is to say the utility of marginal agent i when he adopts proprietary software – is positive or null when $i = 0$. In a similar way, the situation according to which all the agents adopt open source software is always shown to be a solution when u_i^{OS} – that is to say, the utility of marginal agent i when he adopts open source software – is so that $u_i^{OS} \geq [u_i^P, 0]$ when $i = 1$.

Let us first consider case 1 which is defined by conditions $\alpha_1 (1 - \mu) > \bar{c}^{EU}$, $dq - c_L > -\alpha_1$ and $dq - c_L < \alpha_1 (2\mu - 1) + d\alpha_2$. The equation of the curve which provides the utility of marginal agent i when he adopts proprietary software is $\alpha_1 (1 - i) + dq - c_L$. From figure 1a, one can see that the slope of the u_i^P -curve for marginal agent i does not depend on its type (*i.e.*, developer or end-user). Indeed, such a slope only depends on the level of externality which is generated by proprietary software, that is to say the rank of marginal agent i . This property explains why the shape of u_i^P is linear. The equation which corresponds to the level of utility which is reached out by marginal agent i when he adopts open source software is $\alpha_1 i + \alpha_2 \frac{1}{\mu} d$ if $i \in [0, \mu]$ and $\alpha_1 i + \alpha_2 d - \frac{i-\mu}{1-\mu} \bar{c}^{EU}$ if $i \in [\mu, 1]$. As a consequence, the slope of u_i^{OS} depends on the rank of marginal agent i . Indeed, this slope is $\alpha_1 + \alpha_2 \frac{1}{\mu} d$ if $i \in [0, \mu]$ and $\alpha_1 - \frac{1}{1-\mu} \bar{c}^{EU}$ if $i \in [\mu, 1]$. According to the sign of this last term, we can distinguish two types of cases, namely i) cases in which adoption costs only dampen the effect of adoption externalities and ii) cases in which adoption costs are higher-level than the adoption externalities that are supported by end-users. Curves u_i^P and u_i^{OS} may exhibit no, one or two intersection(s) within the interval $[0, 1]$. Let us first consider the case in which one intersection occurs. In this case, this single

intersection between curves u_i^P and u_i^{OS} corresponds to the critical level of adoption of open source software which is sufficient eventually to lead to the adoption of open source software. By supposing that all the agents j ($j \in [0, i]$) do adopt open source software, this single intersection defines the i^{th} agent who indifferently adopts open source software or open source software. Subsequent agents then prefer to adopt proprietary software.

Figure 1b presents the size of the population of users who adopt open source software as a function of its expected level. From A2, we have $u_i^{OS} \geq u_i^{\emptyset} = 0$, $\forall i$. A2 hence rejects that $\tilde{n}^{\emptyset} \neq 0$ for any case. Hereby, $\tilde{n}^P = 1 - \tilde{n}^{OS}$ and transformation $n^{OS} = t^{OS}(\tilde{n}^P, \tilde{n}^{OS})$ can be expressed as a single-variable function of $\tilde{n}^{OS}$. Figure 1b thus represents a transformation of $n^{OS} = t^{OS}(\tilde{n}^P, \tilde{n}^{OS})$. From A3, agents adopt open source software in such a way that the end-users with the lowest levels of adoption costs adopt first. The intersections of this function with the 45°-curve can be seen as the levels which are associated with the size of the population of users who adopt open source software that are both expected and realized according to their rational choices. The positions of these interstections define the Nash equilibria of the second-step subgame.

Figures 1a and 1b can be interpreted as follows. As long as marginal user i does not intend to adopt open source software whereas agents j ($j \in [0, i]$) are expected to adopt it, the function that associates the level of open source software adoption with the expected one remains below the 45°-curve. The 45°-curve and a vertical axis intersect so that the ensuing intersection between u_i^P and u_i^{OS} is reached on the increasing part of function u_i^{OS} for $i \in [0, \mu]$. This intersection defines a mixed Nash equilibrium, namely E3. Here, the agents whose adoption costs are lower adopt open source software at E3 whereas the other agents adopt proprietary software. For values of i defined so that $i \in [\mu, 1]$, let us note that the slope of function u_i^{OS} remains positive. There consequently exists a second value for i so that all the agents adopt open source software when the size of the population of open source users is expected to be at least equal to level i . In this case, the shape of the curve that connects the level of adoption of open source software to the expected one is presented in figure 1b. There is a Nash equilibrium, namely E2, which stands at the north-east corner of the box. When no agent is supposed to adopt open source software, the Nash equilibrium E1 of Proposition 2 stands at the south-west corner of the box.

In the case of static expectations, the local stability of the Nash equilibria can also be analysed by using figure 1b. Let us thus suppose that agents expect the size of the population of open source users to take a value between the values which are associated with equilibrium E1 and equilibrium E3. Here, all the agents decide to adopt proprietary software. We hence conclude that all the initial expectations for fairly low levels of open source software adoption rapidly converge to the Nash equilibrium according to which the open source com-

Nash equilibria of the second step of the game : subcases

	Number of equilibria	Number of stable equilibria	Type of stable equilibria	Conditions
Subcase 1	3	2	WTA – Proprietary	$\left\{ \begin{array}{l} \alpha_i(1-\mu) > \bar{c}^{EU} \\ dq - c_i > -\alpha_i \end{array} \right.$
			WTA – Open source	$\left\{ \begin{array}{l} dq - c_i < \alpha_i(2\mu - 1) + d\alpha_i \\ dq - c_i < \alpha_i + d\alpha_i - \bar{c}^{EU} \end{array} \right.$
Subcase 2	3	2	WTA – Proprietary	$\left\{ \begin{array}{l} \alpha_i(1-\mu) > \bar{c}^{EU} \\ dq - c_i > \alpha_i(2\mu - 1) + d\alpha_i \end{array} \right.$
			WTA – Open source	$\left\{ \begin{array}{l} dq - c_i < \alpha_i + d\alpha_i - \bar{c}^{EU} \end{array} \right.$
Subcase 3	1	1	WTA – Proprietary	$\left\{ \begin{array}{l} \alpha_i(1-\mu) > \bar{c}^{EU} \\ dq - c_i > \alpha_i + d\alpha_i - \bar{c}^{EU} \end{array} \right.$
Subcase 4	3	2	WTA – Proprietary	$\left\{ \begin{array}{l} \alpha_i(1-\mu) < \bar{c}^{EU} \\ dq - c_i > -\alpha_i \end{array} \right.$
			WTA – Open source	$\left\{ \begin{array}{l} dq - c_i < \alpha_i(2\mu - 1) + d\alpha_i \\ dq - c_i < \alpha_i + d\alpha_i - \bar{c}^{EU} \end{array} \right.$
Subcase 5	3	2	WTA – Proprietary	$\left\{ \begin{array}{l} \alpha_i(1-\mu) < \bar{c}^{EU} \\ dq - c_i > \alpha_i(2\mu - 1) + d\alpha_i \end{array} \right.$
			WTA – Open source	$\left\{ \begin{array}{l} dq - c_i < \alpha_i + d\alpha_i - \bar{c}^{EU} \end{array} \right.$
Subcase 6	1	1	WTA – Proprietary	$\left\{ \begin{array}{l} \alpha_i(1-\mu) < \bar{c}^{EU} \\ dq - c_i > \alpha_i(2\mu - 1) + d\alpha_i \\ dq - c_i > \alpha_i + d\alpha_i - \bar{c}^{EU} \end{array} \right.$
Subcase 7	3	2	WTA – Proprietary	$\left\{ \begin{array}{l} \alpha_i(1-\mu) < \bar{c}^{EU} \\ dq - c_i > -\alpha_i \end{array} \right.$
			Mixed – Proprietary and Open source	$\left\{ \begin{array}{l} dq - c_i < \alpha_i(2\mu - 1) + d\alpha_i \\ dq - c_i > \alpha_i + d\alpha_i - \bar{c}^{EU} \end{array} \right.$

munity is fully crowded out, namely E1. Let us now suppose that agents expect the size of the population of open source users to take a value between the values associated with equilibrium E3 and equilibrium E2. Here, the agents who declare their intention to adopt open source software are more numerous than expected. Indeed, some of the agents who were initially likely to adopt proprietary software change their strategy and switch to open source software. Successive iterations show that the number of open source software adopters increases until eventually reaching equilibrium E2.

Other cases are shown to apply, depending on the values of both parameters and variables. They are depicted in figures 2 to 7. Let us note that A1 excludes any case in which curve u_i^P has no positive intersection with the ordinate axis, except – from A2 – the case in which there exists one single equilibrium according to which no users adopt software.

*Nash equilibria of the second step
of the game : subcases 4 to 6*

*Nash equilibria of the
second step
of the game : subcase 7*

.../...

APPENDIX 2: NASH EQUILIBRIA OF THE FIRST-STEP SUBGAME

Proof of Proposition 3: Whatever $(\alpha_1, \alpha_2, d, \mu, \bar{c}^{EU}, \beta)$ is, when $c_L \rightarrow +\infty$, and since values for quality q cannot exceed value $q = 1$, demand for proprietary software vanishes and profit is negative. If a maximal solution for expression (4) exists, this solution is always reached for finite values of (q, c_L) . Expression (4) directly depends on (q, c_L) from a continuous way. Besides, it also directly depends on $n^{P^{\dagger\dagger}}(c_L, q)$, and it indirectly depends on (q, c_L) from $p^{\dagger\dagger} = (1 - n^{P^{\dagger}})$. The shape of expression (4) suggests that – when the second-step subgame provides three equilibria – the continuity of the two terms varying with (q, c_L) is a sufficient condition to provide the continuity of expression (4) on (q, c_L) .

The equilibria of the second-step subgame are the fixed points which are related to correspondence τ . Correspondence τ is defined in the proof of Proposition 1 (see Appendix 1). τ is generated by the intersection between $\alpha_1 \tilde{n}^P + dq - c_L$ and $\alpha_1 \tilde{n}^{OS} + \alpha_2 \left(\frac{\tilde{n}_d^{OS}}{\mu}\right) d - c_i^A$, $i \in [0, 1]$. Note that only the first term continuously depends on (q, c_L) and that the intersection then continuously moves on (q, c_L) . Correspondence τ is thus continuous on (q, c_L) and its fixed points are also continuous on (q, c_L) . We can deduce that expression (4) is also continuous on (q, c_L) . Since value 0 is the lower bound for profit, (q, c_L) is defined on a compact and τ admits at least one maximum on this compact, which proves Proposition 3.

An illustration of the continuity of $n^{P^{\dagger}}(c_L, q)$ and $n^{P^{\dagger\dagger}}(c_L, q)$ is given by considering the cases analysed at the end of Appendix 1. Consider for instance the values of $(\alpha_1, \alpha_2, d, \mu, \bar{c}^{EU})$ and (q, c_L) , which are defined by the conditions of cases 4 and 5. Conditions $\alpha_1(1 - \mu) < \bar{c}^{EU}$ and $dq - c_L < \alpha_1 + d\alpha_2 - \bar{c}^{EU}$ are common conditions for these two cases. The switch from case 4 to case 5 corresponds to the transition from condition $-\alpha_1 < dq - c_L < \alpha_1(2\mu - 1) + d\alpha_2$ to condition $\alpha_1(2\mu - 1) + d\alpha_2 < dq - c_L$. Note that the transition case $\alpha_1(2\mu - 1) + d\alpha_2 = dq - c_L$ is not represented. It appears that the correspondence captured by figures 4b and 5b moves continuously by exhibiting a progressive shift of $n^{P^{\dagger}}(c_L, q)$ from the left side to the right side of μ . Similar continuous transitions can be analysed, for example, from case 1 to case 2 and from case 4 to case 7. In the latter situation (switch from cases 4 to 7), one may observe a continuous shift of $n^{P^{\dagger}}(c_L, q)$ and $n^{P^{\dagger\dagger}}(c_L, q)$ with (q, c_L) .

APPENDIX 3: MARKET FAILURES AND WELFARE PROPERTIES

Proof of Proposition 4: Let us suppose that the optimal quality/price (q^*, c_L^*) leads to the case in which two stable equilibria occur, namely $E1 = \{n^{P^S} = 1,$

$n_D^{OS*} = 0, n^{OS*} = 0\}$ and $E2 = \{n^{P**} = 0, n_D^{OS**} = \mu, n^{OS**} = 1\}$. At equilibrium E1, individual utility of users is uniform and given by $u_i^P = \alpha_1 + dq^* - c_L^*$. At equilibrium E2, utility is $u_D^{OS} = \alpha_1 + \alpha_2 d$ for developers and $u_i^{OS} = \alpha_1 + \alpha_2 d - \bar{c}^{EU}$ for the end-users. On the developers' side, the result of the comparison between the two equilibria is so such that E1 is preferred or indifferent to E2 when $\alpha_2 \leq q^* - (c_L^*/d)$ and E2 preferred or indifferent to E1 when $\alpha_2 \geq q^* - (c_L^*/d)$. If (q^*, c_L^*) is an optimal solution for the firm, the expected profit is positive or null and $u_i^P = \alpha_1 + dq^* - c_L^* \geq 0$. On the end-users' side, the result of the comparison between the two equilibria is such that E1 is preferred or indifferent to E2 when $\alpha_2 \leq q^* - \frac{(c_L^* - \bar{c}_i^{EU})}{d}$ and E2 is preferred or indifferent to E1 when $\alpha_2 \geq q^* - \frac{(c_L^* - \bar{c}_i^{EU})}{d}$. One can conclude that WTA – proprietary – equilibrium E1

is preferred by all users and the firm when $\alpha_2 \leq q^* - (c_L^*/d)$, and that a conflict of interest occurs in the other cases.

Proof of Proposition 5: Let us suppose that optimal quality/price (q^*, c_L^*) leads to E1 is such that E1 is preferred or indifferent to E3 = $\{n^{P**} = 0, n_D^{OS**} = \mu, n^{OS**} = \eta\}$, in a framework in which the linear-quadratic specification is defined so that $\mu < \eta < 1$. At equilibrium E1, individual utility of users is uniform and given by $u_i^P = \alpha_1 + dq^* - c_L^*$. At equilibrium E3, utility is $u_D^{OS} = \alpha_1 \eta + \alpha_2 d$ for developers, $u_i^{OS} = \alpha_1 \eta + \alpha_2 d - \frac{(i-\mu)}{(1-\mu)} \bar{c}_i^{EU}$ ($\mu < i \leq \eta$) for the first set of end-users, and $u_i^P = \alpha_1 (1 - \eta) + dq^* - c_L^*$ ($\eta \leq i \leq 1$) for the second set of end-users. On the developers' side, the comparison between the two equilibria is such that E1 is preferred or indifferent to E3 when $\alpha_2 \leq q^* + \frac{\alpha_1 (1 - \eta)}{d} - \frac{c_L^*}{d}$ and E3 preferred or indifferent to E1 for developers when $\alpha_2 \geq q^* + \frac{\alpha_1 (1 - \eta)}{d} - \frac{c_L^*}{d}$. For the first set of end-users (i.e., end-users whose i is defined so that $\mu < i \leq \eta$), the comparison between the two equilibria leads to E1 preferred or indifferent E3 when $\alpha_2 \leq q^* + \frac{\alpha_1 (1 - \eta)}{d} - \frac{[c_L^* - \frac{(i-\mu)}{(1-\mu)} \bar{c}_i^{EU}]}{d}$ and E3 preferred or indifferent E1 when $\alpha_2 \geq q^* + \frac{\alpha_1 (1 - \eta)}{d} - \frac{[c_L^* - \frac{(i-\mu)}{(1-\mu)} \bar{c}_i^{EU}]}{d}$.

For the second set of end-users (i.e., end-users whose i is defined so that $\eta \leq i \leq 1$), the comparison between the two equilibria is always such that E1 is preferred or indifferent to E3.

To sum up, a coordination failure occurs when $\alpha_2 \leq q^* + \frac{\alpha_1 (1 - \eta)}{d} - \frac{c_L^*}{d}$ inasmuch as developers and some end-users prefer E1 whereas the firm and the remaining end-users prefer E3.

Proof of Proposition 6: Let us consider the pair $(\bar{q}, \bar{c}_L)$ which corresponds to the optimal quality/price solution when users have no other alternative than adopting proprietary software or adopting open source software. The solution is defined so that the firm receives the whole surplus and the utility of users vanishes, *i.e.*, $\mu_i^P = \alpha_1 + d\bar{q} - \bar{c}_L = 0, \forall i$. Profit is then $\pi = \bar{c}_L - \beta\bar{q}^2$. Let us introduce (q^*, c_L^*) which represents the optimal quality/price solution according to which users can choose between proprietary and open source software, as well as $\pi(E1) = c_L^* - \beta q^{*2}$ which is the profit of the firm in the case where E1 is the second-step Nash equilibrium.

Let us suppose that $\pi(E1) = c_L^* - \beta q^{*2} = \pi = \bar{c}_L - \beta\bar{q}^2$. Hence, $c_L^* = \bar{c}_L$, $q^* = \bar{q}$ and $u_i^P(E1) = \alpha_1 + dq^* - c_L^* = \alpha_1 + d\bar{q} - \bar{c}_L = 0, \forall i$. Let us consider the agent whose i is defined so that $i = 0$ (*i.e.*, the first developer). At equilibrium E1, the utility of this agent is $u_i^{OS}(E1) = 0$ if she adopts open source software in such a way that she is indifferent between the two types of software. For levels of expected amount of $\tilde{n}^{OS}$ which are higher than 0, her utility is $u_{i=0}^{OS} = \alpha_1 \tilde{n}^{OS} + \alpha_2 \left(\frac{\tilde{n}_D^{OS}}{\mu} \right) d > 0$, with $\tilde{n}_D^{OS} = \inf[\tilde{n}^{OS}, \mu]$. One can conclude that the agent whose i is so that $i = 0$ eventually adopts open source software rather than proprietary software, except in the case in which $\tilde{n}_D^{OS} = 0$. As a consequence, $(\bar{q}, \bar{c}_L)$ cannot be an equilibrium solution, inasmuch as E1 cannot be a locally stable equilibrium if $(q^*, c_L^*) = (\bar{q}, \bar{c}_L)$ and $\pi^e(\bar{q}, \bar{c}_L) = 0$. E1 is always defined so that $u_i^P(E1) = \alpha_1 + dq^* - c_L^* > \alpha_1 + d\bar{q} - \bar{c}_L = 0 \forall i$, which demonstrates Proposition 6.