

# Understanding the Nature and Production Model of Hybrid Free and Open Source Systems: the Case of Varnish

Zegaye Seifu Wubishet  
University of Oslo, Informatics Department  
[zegayews@ifi.uio.no](mailto:zegayews@ifi.uio.no)

## Abstract

*This is a detailed interpretive case study analysis of an open source software project, called Varnish. The conceptual framework is based on the literature covering issues of commons based production models and the organization of open source projects. The comparative analysis reveals that Varnish is a hybrid project, encompassing the features of open source software while managed by a company as a proprietary project would. It is also hybrid in the sense that it employs a combination of hierarchical and commons based peer production model features. This mix of characters addresses a variety of problems related to each of the aforementioned categories.*

## 1. Introduction

Free and open source software (FOSS) is a phenomenon that has been the focus of research drawing from a variety of areas. Researchers in various disciplines ranging from computer sciences and software engineering to sociology, law, politics, management, psychology etc are all interested in dealing with their respective concerns of this interdisciplinary phenomenon.

The production mechanism in particular, is of profound interest to economists and lawyers like Benkler, as it manifests a new organizational paradigm of information and cultural goods. Benkler called this new model of production - Commons Based Peer Production (CBPP) - and argued that they are inherently different from the classical models: market and hierarchy [1, 3]. This new economic model approached from an open source software development perspective, may also appear in other areas of production, such as content, physical goods, services etc

Varnish is an open source software/project, under the management and ownership of an open source software development and consulting company called

Linpro. It was originally designed and developed by a couple of developers and the idea first came from end users. Currently it has got a fairly large community size which participates in the development process. Goldman [7] referred to such company based FOSSs as hybrid systems displaying commonality between pure community based open sources and proprietary systems.

Studying how the production approach of such company based FOSS projects relate to or deviate from the CBPP model and investigating if they display different, perhaps, unique production behaviour is important. Along this line, it is also relevant to understand the nature of such hybrid systems in relation to pure FOSSs and closed source projects.

This piece of work, therefore, investigates the nature of Varnish and analyzes the properties of the underlying production model in comparison with (a) Benkler's CBPP and (b) other classical models (hierarchy and market models). Theoretically, the study contributes in understanding the constituting character of such hybridized projects which appear with increasing frequency these days. Practically, it unfolds the uniqueness and benefits of the Varnish software which is very useful to the stakeholders. Such practical case analysis is so far less attended in similar literature.

## 2. Methodology

### 2.1. General approach

This paper has primarily followed a qualitative case study with interpretive approach. According to Yin [14], this kind of case study is an empirical inquiry which investigates a contemporary phenomenon with in its real life context. The Interpretive approach was adopted as it contributes to a better understanding of the phenomena under investigation through the meanings that people assign to them within a specific situated context [11].

Linpro is a prominent open source developing and consulting company in Norway and they consider Varnish as their show case. The main programmer is a Danish FreeBSD developer and currently he works on part time bases from his home country. There is one full time developer, who is responsible for the community and leading the project. Both of them were taken as respondents in this study. In addition, all the users in the community were also target of the study.

One other project manager/coordinator is responsible for the administration of resources, a sales person deals with users and sponsors, and a team of individuals with Varnish competence at Linpro give support services to customers. These were not part of the study because all the required data was obtained from the two main developers, the users and the secondary sources. It was judgmentally decided that no additional information could be obtained from them in line with the purpose of this study. Besides, the coordinator was newly assigned and was not quite familiar to the situation by the time this research was being conducted. All the permanent workers, however, line up in the organizational structure of Linpro.

## 2.2. Data collection tools

The data for the study was obtained through three sets of techniques: interviews (preliminary unstructured interview and semi-structured interviews), online questionnaires through e-mail, and recorded interview analysis as well as various corporate documents.

The preliminary interview was conducted with the project leader for about 50 minutes, to get an overview of the project. Then more detailed semi structured interviews were conducted with the former for 2 and half hours and with one of the users/contributors for about 1 and half hours. From the user community, only this person responded through interview for reasons of accessibility and also because he was notably mentioned by the developers themselves as very active user.

Online questionnaire was used both as part of the email itself (a kind of e-mail questionnaires) and as an attachment to e-mails. The main developer resides in Denmark so a questionnaire with open ended questions was prepared and sent as attachment to him. A questionnaire was also sent to the Varnish community through the three mailing lists (Varnish-dev, Varnish-bug and Varnish-misc). This questionnaire had few questions, focussing mainly on the motivation and contribution of the users/contributors.

Document-based data sources like presentation slides, wiki pages and related literatures were consulted to get support data. In addition, a recorded

interview of the main programmer with a BSDTalk program was transcribed and used here. It was a 37 minutes long (12, 734 kb) interview and majority of the issues raised were relevant for this study.

## 2.3. Analysis

The analysis technique used here is more or less similar to what is often referred to as content analysis [9]. First, thorough reading and re-reading of the transcription text was done. The recorded interview of the main programmer was also listened several times. Then themes/topics were identified based on issues in the guiding interview questions. The responses of all individuals under each respective topic were then brought together and rewritten in the form of narrative case description to suite the purpose. Next an analytical frame work was prepared from literature on the characteristic features of different production models and organization of open source. Then the data was analysed against these frameworks and discussions were made. Tables are also used to precisely illustrate related issues.

## 3. Analytical concepts/Frameworks

Here important concepts from literature that helped to make the analysis of the data are presented.

### 3.1. Commons based peer production (CBPP)

CBPP as described by Benkler is a socio-economic system of production among large groups of individuals with the purpose of collaborating effectively to provide knowledge or cultural goods. He argued that this model of organizing production is different from the two classical models: managerial hierarchies and market models and possess its own unique characteristic features. It is a new model with out traditional hierarchal organization or financial compensations. People in CBPP participate in a wide range of intrinsic and self-interested reasons. He abstracted the organizational structure of FLOSS projects by the CBPP model and extends it to other areas too.

There are two mentioned economic advantages of the peer production model. The first is the information gain advantage which is achieved by ensuring that individual workers are self selected. Markets attempt to help individuals to decide what they should by assigning price signals to and firms do it by attaching managerial weights. But these need accurate specification of agents, efforts, or resources, which obviously is unattainable. CBPP, however, improve

this problem as it places the point of decision about assigning any given person to any given set of resources with the individual.

The second is the allocation gain which is indicated by the variability in fit of people to projects and existing resources. It is argued that human creativity cannot be easily measured to assume suitability for the job and people can be fit to many multiple jobs. Peer production has an advantage over firms and markets because it allows large groups of individuals to scour larger groups of resources in search of materials, projects, collaborations, and combinations than is possible for firms or individuals who function in markets. The advantages of peer production are, then,

improved identification and allocation of human creativity.

There are, on the other hand, two major problems/challenges associated with the peer production model: motivation and organization. According to Benkler, these relates to the traditional objections of the ‘commons’. The first is that no one will invest in a project if they cannot appropriate its benefits i.e. motivation will lack. And second, no one has the power to organize collaborations i.e. organization will lack and organization will fail.

Table 1 describes the key characteristic features of CBPP and their descriptions. These features are discussed by Benkler as preconditions for a project to qualify to CBPP [1, 2, and 3].

**Table 1. Qualifications/Characteristic features of CBPP**

No	Feature	Description	Type
1	Purpose	The purpose of CBPP systems basically is collaboration	General
2	Coordinating Mechanisms	Relies neither on market pricing nor managerial hierarchies to coordinate the common enterprises – rather it uses social cues and motivations	
3	Areas of application	Production of knowledge or cultural goods - which are known to be non rivalry, presence of excess capacity and ensure low cost of contribution	
4	Infrastructure	digitally networked environment facilitated by the technical infrastructure of the Internet	
5	Modularity	The project/object must be decomposable into modules that can independently be handled	Structural
6	Granularity	The modules should be fine grained i.e. each should be with smaller size.	
7	Heterogeneity	Accommodating more differentiated and variously sized contributions i.e. heterogeneous granularity	
8	Decentralization	The contributors may be geographically/physically dispersed	Nature of Peer
9	Excess Capacity	Abundance of unused capacity and large number of contributors	
10	Self-Selection	Self-assigned contributors as opposed to managerial assignments.	
11	Collecting and Storing	Mechanisms of gathering the proposed/initiated ideas from all the different sources	Integration and product
12	Redundancy	As a mechanism of correcting errors and ensuring if modules are covered	
13	Editing and incorporation	Quality controls (filtering mechanisms) and low cost integration mechanisms	
14	The commons	In the end a certain type of common product will be produced for use by anyone in need	

### 3.2. Firms and market models

The market based model is identified as tagging different prices to different jobs which serves as an attractor to anyone interested in doing the job. Prominent economists describe markets as one very important mechanism for solving the problems of coordination and motivation that arise with interdependencies of specialisation and the division of labour. Market institutions leave individuals free to pursue self-interested behaviour, but guide their choices by the prices they pay and receive. The

decision is influenced by competition, supply and demand [19].

Williamson [13] argued that transactions that involve uncertainty about their outcome, that recur frequently and require substantial ‘transaction-specific investments’ – of money, time or energy that cannot be easily transferred – are more likely to take place within hierarchically organized firms. Exchanges that are straight forward, non repetitive and require no transaction-specific investments will take place across a market interface. The firm production is identified as a centralized decision process determining what has to be done and by whom.

Powell [8] wrote about ‘network’ form of organization as distinct from market or hierarchy. He identified a coherent set of factors that make it meaningful to talk about networks as a distinctive form of coordinating economic activity.

He made a stylized comparison of forms of economic organizations based on some criteria (discussed in table 4). These criteria are used in this study to indicate how the Varnish project relates to the three models (the third being Network model, according to Powell).

### 3.3. Open source vs. commercial projects

Open source software development project, according to Krogh and Hippel [10], is typically initiated by an individual or a small group for an intellectual or personal or business reason. Such projects are generally Internet-based networks and follow community intensive approaches to the development [12]. The communications are openly accessible and publicly available over the Web [12] and the software they develop is made freely available to all that adhere to the licensing terms specified by the open source project. The number of developers participating in a project ranges from a few to many thousands, and the number of users of the software produced by such projects range from few to many millions [10].

In commercial software practice, the development of software products tends to hang on tight management of the processes: software firms regulate the relationship with their programmers through carefully drafted employment contracts, divide labour and allocate work responsibilities along the development process, and take precautions to prevent employees from leaking software-related trade secrets and information to competitors. For-profit programming firms seek to reduce development costs and control quality by closely monitoring what programmers do and how they do it [10].

Proponents of FOSS and researchers identify as the main advantages of FOSS including cost reduction, availability of source code, the freedom (to copy, use, study, improve, redistribute) and the benefit of directly involving users or any interested person to take part in the development [5]. Fitzgerald [6] argued that the technological motivation for OSS development directly relates to the software crisis, which clearly illustrates that traditional modes of development do not work very well, specifically in the areas of speed, quality, and cost of development.

On the other hand, much of the critical FOSS literature raise different issues as weaknesses of the FOSS approach or as advantages of the traditional

proprietary approach. Some of these problems relate to absence of a help desk support, reliability, accountability, lack of awareness by the larger community due to installed base problem, shortage of trained man power in many places etc. [5]

The hybrid model, however, appears to benefit from both the community and firm models. The community offers an almost boundless space for experimental projects and interactions between diverse actors, while firms stabilise and standardise the FLOSS-based products by incorporating them together and distributing in the market or by supporting related services [7].

The new hybrid model has been spurred by the entry of companies and software houses producing free software, giving it away to customers free of charge and shifting the value from licensing agreements to additional services such as packaging, consultancy, maintenance, updating and training [4].

## 4. Description of the Case

The flow of presentation here is based on identified themes/topics.

### 4.1. What is varnish?

Varnish is a state-of-the-art, high-performance HTTP accelerator. Most web sites or content management systems (CMS)<sup>1</sup> present dynamic web pages consisting of a number of different elements. Combining these elements is both time-consuming and CPU intensive, and the process is repeated for every individual user, even though the content is often identical.

Varnish temporarily stores the most frequently requested pages in cache memory and effectively presents these pages from cache. It is a tool designed solely to do web-acceleration for slow CMS systems and it is built for server side caching. It is not designed for websites that have only small objects.

Respondents said that it has got better configuration, better management, much faster and with content management features than the available similar systems like Squid<sup>2</sup>. It is also documented that

---

<sup>1</sup> A CMS is a computer software system used to assist its users in the process of content management; facilitating the organization, control, and publication of a large body of documents and other content, such as images and multimedia resources.

<sup>2</sup> Squid is software that caches Internet data. It does this by accepting requests for objects that people want to download and handling their requests in their place.

it has got high performance with modern features like Virtual Memory, multi-processor, multi-thread (and every other trick in the book).

Describing it in short, the main programmer, stated that "...Varnish is a cache put in front of your web server if you have performance problems and apparently performance problems is something pretty much every body has so uses CMS of any kind."

It was targeted primarily at the FreeBSD 6/7 and Linux 2.6 platforms. Now it is portable and works on FreeBSD, Linux, Mac OS X, other BSDs and Solaris.

#### 4.2. How was the project started?

It was started in early 2006 by the request of the Verdens Gang (VG) company running a news website (VG.no), which is one of Norway's largest in terms of traffic. The idea was originally initiated by server managers of VG and Aftenposten (another popular newspaper company in Norway). Earlier they were using CacheFlow appliances to their servers. But CacheFlow was not so fast and users were usually complaining about it. Then they started working with Squid and were able to reduce the number of servers required to a certain extent. But as the traffic grows bigger and bigger every time, so they needed more servers and faster servers at the same time. And the performance of Squid was found out to be poor to cope with that.

These individuals discussed their shared problem – in their eyes at that time, they had many Squid servers whose performance and flexibility as a web accelerator was poor. They were unhappy with that performance and stability – slow response, terrible peak-handling and significant loss.

So there was a dire need for better solutions and there was no such open source software for this task at that time. The options available were using Apache, Squid or proprietary products. Apache is not built for caching purpose – it is something that apache not good at or they call it 'a square peg in a round hole' for the job. The other option was dealing with closed source products which will not give close control and results in getting locked in their service. Squid, the third option, is built for client side caching, which is known to be troublesome. It was also known to be fragile, buggy, has got lots of unwanted functionalities (like authentication), wrong configuration features.

Varnish was then born - designed and coded by the Danish programmer during its first version. This first version (version 1.0) was sponsored by VG with 500.000 Kr (\$85K) and the program had 22000 lines of code.

The resulting status of using Varnish now is a more than ten fold increase of performance compared to

competitors, and a major reduction in hardware requirements. For instance, VG now is able to run its entire web site on one server, where they previously needed 12 Squid machines (servers). Aftenposten, now use Varnish as a web accelerator for its website (www.aftenposten.no) and to their sister organizations too. The result is that, they use 3 Varnish servers for Aftenposten which was 8 Squid servers earlier.

#### 4.3. Why Varnish is decided to be open source?

From the beginning, the people at VG persistently asked it to be open source for two reasons. First they logically demanded continuity. They knew that this is possible only if they get the source code, regardless of the fate of Linpro or changing relationships. Second, they use many other open source programs including FreeBSD and so they wanted Varnish to be of that type.

Replying to the question why Linpro accepted the request to be in open source, the project leader said that Linpro is originally committed to promote the use of FOSSs – "FOSS is kind of alpha and omega for us."

The main programmer expressed his firm commitment to open source by relating his bad experience in that "because I was sitting at 3am, trying to start a process that had died only to see it die again immediately, and lacking source and a vendor who didn't answer the phone. What could I do? I was helpless." Now, for him FOSS is a precondition to be involved in any such work and stressed that he has always been pushing people to use open source.

#### 4.4. Organization of Varnish

The Varnish project, according to the project leader, can be classified as a kind of company based FOSSs, which have clear road map and goal of their own and perhaps a strategic plan of the owning company. Here the developers are hired and paid for their work and the company is making money out of it. They also face tighter control and scheduled deliveries.

Speaking about the first Version of the software, the main programmer said that "Varnish is to a large degree a one-man-show since he has written quite close to 100% of all the code. It should be compared with a similar project, like to the very early Linux, back when that was essentially also a one-man-show." He believes that this is normal for a first version of something and once the platform is there, it is easier to get more people involved, and they certainly hope that will happen down the road. Both developers indicated that more contributors are showing up now and feed back is also coming from customers.

Many of the external contributors so far participate in bug fixing, bug reporting and interact in question and answers.

Unlike the case of many other FOSSs, it is a customer (the VG Company) who initiated the development of Varnish. They went to Linpro because they wanted it to be open source and Linpro is known to be dedicated for developing FOSSs principally (VG already had previous links for other services with Linpro). In addition, the people at VG knew that they were not in software development business so they wanted a committed partner which can ensure updates for new features and new developments. They knew that they will not do these themselves.

In terms of administration (like on funding, controlling, planning and staffing), Varnish and other hybrid FOSSs appear to be similar to proprietary projects. The owning company handles many of these activities.

#### 4.5. Licence type and business strategy

Varnish is released under the BSD license. The reason given by the project leader to this is that VG was using free BSD earlier and that was their preference. But he also recalled the argument raised at that time - "GPL give freedom to the code where as BSD give freedom to the client. The software has no feeling; it is people's freedom that should matter."

According to the main programmer, BSD actually was pretty much a requirement of VG from the beginning. But he didn't protest it because he was firmly supporting the BSD license instead of other licenses. However, he related his speculation about the decision to that one of the reasons could be that Varnish has a large potential for going into a high profit commercial area, basically competing with big proprietary businesses. Thus, using the GNU license would probably prevent people from even thinking about that. So he suspected that could have been part of the reasons taken for choosing the BSD license.

Linpro makes money from the customers through sponsorship, for the additional features that they may demand and for the support services – like installation and configuration of the software, support, and hosting. Some features are unique to each company so the system has to be customized and expanded to accommodate that.

Sponsorship is also one major source of fund. The project's trac wiki holds that there are three current sponsors – Aftenposten (Norway's largest subscription newspaper), API (A-pressen Interaktiv, Norway's largest digital media network) and Escenic (an online content management systems provider to European

media companies). VG, which initiated the project, is not an active sponsor now.

#### 4.6. Tools used

Subversion is one of the open source tools being used by the project for version controlling and management. TRAC, a group ware with a wiki is used for the project management and bug tracking. It allows them hyper linking information between bugs, subversion and wiki content. It also serves as a web interface to subversion. Mailman is used for mail handling with its features of list administration, archiving of messages and spam filtering. GNU author tools are also often used. Varnish is developed entirely by C programming language and at some places Perl is used.

#### 4.7. Structure and Filtering Mechanisms

Varnish can be said is modular in the sense that structured interfaces have been used to demarcate areas that can be replaced with different implementations. For example, there are three implementations of the cache acceptor and two implementations of the storage module so far. These modules are fairly smaller (fine grained) by any outsider look and have small number of lines of code.

The project leader mentioned that there was a time pressure so that modularization was not given much attention by then. But later it was improved and some parts of it are easily pluggable now and have very well defined interfaces but some are not. For instance, it is learnt that the hash algorithm is only one yet. There are also obviously meticulous organizational works in it like the manager/worker split mentioned above.

Regarding the presence of mechanisms to filter malicious contributions or noisy mails, the project leader said that final decisions are made by the two main developers. Any contributor could send patches and that will be evaluated. The main programmer stressed that so far the contributions have not been an excessive load for the leaders, so checking everything going in and out is simple.

#### 4.8. Geographic distribution of the community

From the mailing list it is possible to trace and tell that the Varnish community is comprised of individuals all over the world. The main developer, to start with, works from Denmark on par time bases. The other main developer is a permanent worker at Linpro, Norway. The distribution of those users who responded to the questions through mail was from Norway (3), China (1), New Zealand (1), USA (2) and Brazil (1).

The person who was mentioned as contributing the biggest patch was also from Germany.

#### **4.9. Motivation - Why people engage in Varnish?**

The main programmer first mentioned that there is money involved in Varnish, "I am paid to program it and VG gets paid to do project infrastructure etc. But that is not at odds with open source. Because programmers have to live, feed their kids and keep a roof over their heads just like everybody else." He emphasized that personally for him there was a motivation - the money - but he had other offers on the table which settled it. One such offer was that he has spent 15 years working on and to a large extent renovating the FreeBSD kernel, and Varnish gave him a chance to go out in "user-land" (as kernel coders call it) and apply some of the techniques, features and principles, to make an example of how modern programming should happen.

He stated that he was sceptical when VG approached him with the Varnish Project because that was not really his specialty and not really his preferred area. On the other hand he stressed that he was tired of people writing lousy programs and blaming it on his kernel. So he considered this as a fine chance to educate by example. In fact, he emphasized in his interview that he was highly driven by the motive of teaching others with example of how to do performance programming with the state of the art technology.

Another interesting reason that he presented was that though he was in it for money, it is never that simple with software, the production of which is not unlike artistic expression like sculpting or painting: "Once you've written 10,000 lines of code, it becomes increasingly hard to walk away from your program, so the line between 'for money' and 'for idealistic reasons' gets increasingly blurred as time goes by."

The project leader also responded that "The motivational factors for me are both interest and money. I was very much interested at first when the idea was presented. But I wouldn't have done it if I was not paid because I may not have time to do it for free - the work was quite big and it needed a lot of time and dedication."

One of the often interacting and contributing users of Varnish replied that he was motivated out of interest and by the desire of getting Varnish to be a better solution for his employer (Aftenposten). "To make Varnish successful in the long run", he said, "it is crucial that the project becomes popular because it is not a proprietary product to be taken care of otherwise." He also has the desire to learn more and

get experience on open source software development, as he and his customers use open source systems a lot.

The other users who responded through e-mail raised the following factors as their motivation to participate in the Varnish community: interest and enjoyment to help others in solving problems, interest in the concept of HTTP caching and the need to learn more on it, to get experience on open source development for use in future professional career, to get solutions for problems with respect to Varnish from the mailing list, the need to make varnish robust and popular for ones own use, to see the open source community grow so that innovation expands and continuous, believe in the philosophy that giving back to the project will increase its usefulness to me in the long term, the developers are highly experienced and very approachable making it hard to get away.

As a challenge and/or opportunity to the work, the main programmer indicated that "the biggest problem was that I didn't run a big website myself, so I was not intimately aware of all the 'standard tricks of the trade', lingo and products people talk about." And he admitted that the contribution of the users in familiarizing him with ideas and tricks there was immense. Their domain knowledge about web servers and related stuffs was very important to the success of the system.

#### **4.10. Future plan**

It is mentioned by the project leader that the project has got a road map and ideas - partly written and partly in mind. The website of the project also displays some of them. He stressed that the company's decision every term has its influence.

The main programmer also indicated that they will keep on improving it, as he believes that it is too much of a success to just go away again and there are many ideas for Varnish 2.0. He also remarked that "since this is 'work' for me, the direction will very much depend on whether we can find sponsors or otherwise." They are hopeful that some of these professional CMS houses or some of the big news sites would help them get funds to exert a concerted effort on next version. Or they are also hopeful that may be some volunteers will come up and do it.

Actual customers are also helping the development by sponsoring. Aftenposten, for example, is one of the three potential sponsors and paid around 200,000 kroner, so far to the owning company, Linpro.

### **5. Analysis and discussion**

Based on the topics in the analytical framework above, the analysis and discussion here is presented in four sections.

### 5.1. Varnish and FOSS

First it is relevant to see the nature of Varnish with regard to other FOSSs. Goldman [7] wrote that, in pure open-source projects, all the workers are volunteers. There are rarely if ever formal processes that are followed or formal specifications produced. The team may or may not have usability experts, documentation writers, testers, or project management of any variety.

When companies start open-source projects, it is typical for those projects to be part of (or all of) a company initiative being done for strategic purposes and aims. Because many companies use a more traditional or conventional development process, such companies starting open-source projects are likely to want to engage the open-source community with some of their traditional tools and in-house experts. When this happens, we call it a "hybrid open-source project" [7].

Accordingly, Varnish can be categorized as one type of hybrid system. From the project side it has got a parent/owning company (Linpro) with paid manager, developers and sales person. Like proprietary projects, different administrative works are formally done (planning, controlling, hiring and funding etc). It has got a road map and goal of its own, and which is inline with the parent company. There are also deadlines to be met in some cases and scheduled deliveries. For example, it was learnt that the first version was supposed to be released in three months as per the deal. The continuity of the project also depends on the availability of sponsors and yearly budget of the company. These explain the proprietary face of the project.

From the production side, we find the open source face of it. It has got a reasonably large community of users participating in the use and development. Like many other FOSSs, at first it was mainly developed by a single person and through time it has attracted more people in the community. Currently the mailing list is said to be self sustained in that the members of the community sufficiently interact with each other in the form of questions and answers, with out even the interference of the leaders. Varnish also has got sponsors that support the development of the software.

A unique feature of Varnish is that the customers/users are companies handling complex content management systems. These companies have their own technical people managing their systems. This is advantageous in that the community is comprised of a bit of technical people. It is also limiting in the sense that it couldn't enjoy mass-market as it is not meant to be used by any person that do not want to speed up large CMS or run large web servers.

For Varnish or any other such hybrid systems, it can be said that there are two types of users. One is the clients who directly contract with the company for support or future enhancements or any other task and also they directly pay for those services. In the case of Varnish almost all the active clients are companies with their own facilitators or professionals handling the job at home. Those individuals happen to be part of the user community in the development. The second category is of those users who find the software on the web and make use of it for their own sake, or perhaps for their hiring company, with out any payment to Linpro. These people also become part of the community. In this study, all of them are referred to as the 'users' of Varnish.

**Table 2. Motivation of Stakeholders**

Motivation	Users	Developers	Linpro	Sponsors
Reasons to Take Part	Interest	Money/Salary	Money - making profit out of the product	Benefits – like cost saving, fast services and better solution
	Benefits to their hiring company	demonstrate a model system – prove to others	Having a show case to promote the company	
	Getting experience in open source development	Try out a new task – from kernel development to user land		
	Help others and learn from them			
	Improve the usefulness of the software and increase its popularity			
Reasons to make it Open Source	To gain more knowledge and enrich oneself	Openness is a precondition – due to experienced troubles with closed sources	Commitment to promote and make business out of FOSSs	Continuity – even in the absence of the developers
		Response to the user's and the hiring company's request	Response to the user's request	Compatibility with already existing systems

Since Varnish is hybrid system, the different stakeholders, including the users, have different motivational factors on the project. Table 2 summarises those reasons that make each of them take part in the project and have it open source.

An interesting point here is that having it as open source offers both guarantee of continuity and a reliable partner. Even in the absence of Linpro, the client companies can proceed on their own. VG knew that they are not in software development business so they wanted a committed partner which can ensure updates for new features and new developments, which related them to Linpro.

User companies, even those that are in competing businesses, here are collaborating and sponsoring the development of software that is commonly beneficial to all of them. Such approach can be considered as a good model to the software development industry.

In such arrangements, the motivational factor for open source developers/programmers may be different but money is among them. This actually is never at odds with the idea of FOSSs, as programmers make their living out of it. The hybridization of the project appears to ensure the fulfilment of this basic need of programmers, unlike to what people fear in pure open source projects.

**5.2. Varnish and CBPP**

One core purpose of this study was to understand the production model of Varnish and similar hybrid projects. In view of addressing this, a comparative discussion of how Varnish relates to or qualifies the preconditional features of CBPP is presented in Table 3 below.

**Table 3. Varnish VS CBPP**

No	Type of Feature	Varnish
1	Purpose	The purpose of Varnish is both profit making and at the same time collaboration through the ideology of FOSSs.
2	Coordinating Mechanisms	The coordinating mechanism is a mix of both hierarchy (from the company side) and peer production (from the development and use side)
3	Areas of application	The project, as an open source based software development system, is one potential application area where CBPP could happen.
4	Infrastructure	The development takes place over the internet (online) and the project could be classified as hybrid FOSS.
5	Modularity	Some parts of it are well modularized, easily pluggable and have well designed interfaces; but some parts are not.
6	Granularity	The available modules are fairly smaller in size with reasonably smaller lines of code.
7	Heterogeneity	There is no observable heterogeneous granularity so far.
8	Decentralization	Decentralized community - the developers and users are geographically separated
9	Excess Capacity	Excess capacity is there and being exploited in Varnish - all the community members, including the main developer, are working on Varnish not as full time job. They do it together with their other major works in a way of utilizing excess capacity.
10	Self-Selection	The main developers are assigned by the company but many of the user contributors are self-identified.
11	Collecting and Storing	The mechanisms of gathering information and patches in Varnish mainly are the various mailing lists and the version controlling tool.
12	Redundancy	Redundancy appears to be minor as the communication among the contributors is fast through the channel – allowing to know what each others are doing and what problems are being reported every day.
13	Editing and incorporation	Incorporating patches and final decisions are done by the main developers The developers make sure that wrong patches are not committed (filter) Since version controlling and other open source tools are in use it can be said that the integration cost is very low.
14	The commons	The software is a commons product that anyone can pick it and use

It is well indicated earlier that all the users are geographically distributed all over the world and even the main developer works from his own home country. Their work is highly facilitated by the digital network infrastructure and the different open source tools.

From the source code of the software and the project’s Wiki (like the “ideas list” and the few

suggestions there given for contributors) it is possible to tell that the modules are reasonably smaller, which could be done independently and separately.

Regarding self-selection, Varnish appears to be mixed. The developers are hired and assigned managerially by the company; there is also project manager, sale’s person and facilitators hired. They

all are paid and guided by the policies and procedures of the hiring company. Such is typical of hierarchical organization. On the other hand, the community users can be said are self selected. One main contributor mentioned earlier, for example, just came out of the blue to enable Varnish run on Solaris and send a big chunk of patch – no relation with developers or the company. Other users are participating because they are using Varnish for their own sake and want it grow up more.

**5.3. Varnish and the classical production models**

The analysis here is based on Powell’s style based comparison [8] of forms of economic organizations. The third model that he called it ‘Network’ is almost similar to the Benkler’s CBPP model. All the features and the corresponding comparisons with in the three models, below in Table 4, are taken from his work as they are. Here the additional comparison is done

against the Varnish project features to show the similarities and differences.

This comparison further revealed and strengthened the hybrid nature of Varnish in terms of production style. As it is illustrated in the table, the varnish project appears to share many key features of the Network and Hierarchy model descriptions of the author and very small elements of the market model.

The degree of flexibility of Varnish can be said is medium to high because there is organizational and developer’s/contributor’s interest, which should always be compromised - a two way street where there is little or no dictation by one side.

The preferences of the different actors of Varnish – the developers, the owning company (Linpro) the community members and the user companies – can be said is interdependent because the decisions and choice made by one affects the others to some extent as there is mutual or overlapping interest. But still there is room for independent decisions

**Table 4. Varnish VS the three production models**

<i>Key Features</i>	<i>Market</i>	<i>Hierarchy</i>	<i>Network</i>	<i>Varnish</i>
Normative basis	Contract – property rights	Employment relationship	Complementary strengths	Both employment relationship and complementary strengths
Methods of conflict resolution	Haggling – resort to courts for enforcement	Administrative fiat – supervision	Norm of reciprocity – reputational concerns	Supervision to a certain extent and community norms (may be reputation)
Degree of flexibility	High	Low	Medium	Medium to high
Amount of commitment among the parties	Low	Medium to high	Medium to high	Medium to high (based on responses related to motivation)
Climate	Precision and/or suspicion	Formal, bureaucratic	Open-ended, mutual benefits	Open-ended, mutual benefits, formalities and bureaucracies
Actor preferences or choices	Independent	Dependent	Interdependent	Interdependent
Mixing of forms	Repeat transactions, contracts as hierarchical documents	Informal organization, market like features, profit centre, transfer pricing	Status hierarchies, multiple partners, formal rules	Formal and informal organization, status hierarchies, multiple partners, formal and informal rules

**5.4. The benefits of the mix**

Varnish, partly as a CBPP project, obviously has enjoyed one important economic benefit of this model (over the market and hierarchy) – the information gain. This is prevalent because the community members have different back grounds, varying capacity, experience and motivation.

Another interesting point is that the way Varnish is developed and structured appears to alleviate/solve the

two major problems/challenges associated with the peer production model itself: motivation and organization. According to Benkler, the first relates to lack of motivation to contribute, as in ‘commons’ no one is to appropriate the benefits. This is solved here because there is a company (Linpro) driving the development and owning it. The contributors are also directly using the system to their hiring company, where indirectly they are paid for their job. The second relates to lack of organization. This is obviously solved

in Varnish because the company is doing the coordination and driving the development.

Thus, the Varnish project approach, which appears to have mixed features of CBPP and hierarchy, could solve the two infamous problems of the ‘tragedy of the commons’ (associated with pure CBPP projects) and the information gain problem (associated with market and hierarchy).

In addition, Varnish as a hybrid open source project appears to be enjoying the benefits of both the pure open source model and the proprietary approach having an accountable firm.

## 6. Conclusion

This case study analyzes one exemplary model of hybrid systems that combined features of proprietary and open source approaches as well as that manifested the features of hierarchy and peer production models. This hybridization helped it to get the benefits of all the mix. It also avoids the limitations associated with CBPP and hierarchical production models, while ripping the advantages of the two. The way Varnish is organized and developed is believed to be appropriate to similar domain specific and/or relatively small-user-pool projects. Further research is required in order to identify the ways in which such hybrid models may foster creativity, innovation and learning.

## 7. References

- [1] Benkler, Y. "Coase's Penguin, or Linux and the Nature of the Firm." *Yale Law Journal*, 112:79. 2002.
- [2] Benkler, Y. ""Sharing Nicely": On shareable goods and the emergence of sharing as a modality of economic production." *Yale Law Journal*, 114: 85. 2004.
- [3] Benkler, Y. "Commons-Based Peer Production and Virtue." *Political Philosophy* 14(4): 25. 2006.

[4] Bonaccorsi, A., Giannangeli, S., Rossi, C. "Entry strategies under competing standards: Hybrid business models in the open source software industry." *Management Science* 52(7): 40. 2006.

[5] Feller, J., Fitzgerald B., Hissam, S. and Lakhani, K. "Perspectives on Free and Open Source Software." MIT Press, 2005

[6] Fitzgerald, B. a. F., Joseph. "A Framework Analysis of the Open Source Software Development Paradigm." ICIS '00: Proceedings of the twenty first international conference on Information systems, Association for Information Systems, 2000.

[7] Goldman, R. a. G., Richard P. "Innovation Happens Elsewhere: Open Source as Business Strategy." Morgan Kaufmann, 2005.

[8] Powell, W. W. "Neither market nor hierarchy: Network forms of organization." *Research in Organizational Behaviour* 12: 41. 1990.

[9] Taylor-Powel, E. and Renner, M. "Analyzing Qualitative Data." University of Wisconsin Cooperative Extension. Publication G3658-6. 2003.

[10] von Krogh, G. and von Hippel, E. "Special issue on open source software development." *Research Policy* 32: 9. 2003.

[11] Walsham, G. "Interpretive case studies in IS research: nature and method." *European Journal of Information Systems* 4: 7. 1995.

[12] Walt, S. "Free/open source software development: Recent Research Results and Emerging Opportunities." Proceedings of the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering. ACM, 2007.

[13] Williamson, O. E. "Market and hierarchies: Analysis and antitrust implications." The Free Press, 1975.

[14] Yin, R. K. "Case Study Research Design and Methods." Sage, 2003.