Sciences Po Rennes
Section Écofi
Séminaire Stratégie des entreprises européennes, américaines, et tiers-mondiales
Under the direction of Dr Bernhard Kitous

Google Chrome:
free software as a launching platform

Charles-Axel Dein
ca@d3in.org — charlesaxel.dein@gmail.com

Rennes (France) — May, 2009

Preliminary remarks

This dissertation has been written in May 2009 to fulfil the requirements of a seminar called “Strategy of European, American and Third-World Companies”, at Sciences Po Rennes, *section Écofi*. It has been written in accordance with the principles of the *Sciences Po* education, which includes interdisciplinarity. Accordingly, this paper uses model from law, economy, sociology and history.

French being my mother tongue, please apologise any grammatical or orthographical mistakes. Feel free to contact me (`ca@d3in.org`).

Executive Summary

Google has chosen to release Chrome as Free and Open Source Software (FOSS) in order to achieve two goals.

Protect its businesses. FOSS is used to foster Chrome's market penetration, which will in turn encourage the diffusion of innovation in the web value chain. It is in the firm's interest to make the Internet grow: since it has been able to privatise the Internet, growth means increased audience, which will help Google sell advertisements (97% of its revenues) and gather information (via crowdsourcing and behavioural marketing).

Prepare an offensive in cloud computing. FOSS in Chrome shows that Google aims to be a leader in the cloud computing paradigm. It will be able to license its web-applications and diversify its revenues.

Chrome is FOSS because it will not generate value by itself. Its purpose is to serve as a *launching platform* for innovation in the Internet. These innovations will indirectly generate cash-flows.

Keywords. Google, Chrome, strategy, innovation, open web standards, free and open source software, Internet, browser, cloud computing.

French version

Google a choisi de faire de Chrome un Free and Open Source Software (FOSS) afin d'atteindre deux objectifs.

Protéger ses activités. L'utilisation du modèle de développement FOSS permet d'accroître la pénétration de Chrome dans le marché des navigateurs. Ceci va à son tour accroître la diffusion de l'innovation dans toute la chaine de valeur Internet. Il est dans l'intérêt de Google de faire grandir le web, car la firme a de fait réussi à privatiser Internet, que l'on pourrait pourtant considérer comme un bien public. En conséquence, faire croître le web agrandit son audience, donc les revenus publicitaires (qui représentent déjà 97% de son CA) et facilite la collecte de données qui peut être indirectement monétisée (*crowdsourcing*, ciblage comportemental).

Préparer une offensive dans le *cloud computing*. Faire de Chrome un FOSS montre que Google a de très grandes ambitions dans le marché du *cloud computing*. Il est vrai qu'un tel paradigme permettra à l'entreprise de diversifier ses revenus en vendant des licenses de ses web-applications.

Chrome est donc un FOSS parce qu'il ne générera pas de valeur « par lui-même ». Il est en fait une *rampe de lancement* pour des innovations qui vont indirectement générer des revenus pour la firme de Moutain View.

Mots-clés. Google, Chrome, stratégie, innovation, standards web ouverts, Internet, navigateur, cloud computing.

Contents

List of Figures

List of Tables

Acknowledgements

I am first very grateful to my dissertation chair, Dr Bernhard Kitous, for his continuous support, expertise and challenge to my work. This work would have never been possible without his mentoring and his constructive feedback.

I would like to thank Mr Bregham Dagliesh, for his course about the dynamics of technological innovation, which provided instrumental concepts and models used in this paper.

This paper also strongly relies on many thoughtful interviews conducted with helpful people. Their insights are considerable assets for this work. That includes Mr Pierre Fremaux (FaberNovel), Mr Tristan Nitot (Mozilla Foundation), Mr Richard Stallman (Free Software Foundation), Dr Eben Moglen (Free Software Foundation), Mr Serge Cheminade (Orvinfait), Mr Philipp Lenssen (Blogoscoped), Mr Jérémy Milhau (Zorgloob), Mr Étienne Bersac (Nerim).

Thanks also to my fellow researchers Térence Neu and Henri Lanièce for their insights from their related work about Google's strategy.

A. Case description: Google Chrome in the browser war

I. A BROWSER MARKET STILL DOMINATED BY AN AGEING IE

A *web browser*[1] is an applications used to display web pages (like `www.google.com`) on the Internet. It is now the most important software on a computer, because of the proliferation of highly interactive web applications.

The first browser war resulted in the victory of Microsoft's IE (Internet Explorer), which had 96% market share in 2002, and the defeat of its rival, Netscape Navigator. Subsequently, Netscape released its browser's source code, and fostered the creation of Mozilla Foundation which developed Firefox.

The browser market remains highly concentrated, with only five main actors:

- Internet Explorer (Microsoft), 67% market share [Net Applications, 2009]
- Firefox (Mozilla Foundation), 22%
- Safari (Apple), 8%
- Opera (Opera Software), 1%
- Chrome (Google), 1%

Internet Explorer's historical market predominance has one main reason: it is shipped as the default browser in Microsoft Windows, the dominant operating system (OS). Yet, because of this long monopoly, IE's quality have significantly decreased, especially regarding security and web standards compliance. It has consequently slowed the diffusion of innovation in the Internet sector. Microsoft implemented proprietary features that were unavailable on other platform, which has created a sort of lock-in. It has also fostered wrong programming practices that makes IE-designed websites appear very different or non-functional on other browser. Latest version of IE are more and more standards compliant and introduced many significant features improving its security and the speed of its JavaScript engine, even if in many fields it remains an innovation follower.

Firefox's market share has been steadily increasing since a few years. Its third version has even become the most popular browser in Europe in a breakdown by version number [2]). Whereas Microsoft's browser is pre-included on most PCs, Firefox can be downloaded for free on `www.mozilla.com`. Firefox's user base is mainly composed of innovators and early adopters, while current growth rates show wider market acceptance. Despite its smaller market share, most specialists reckon that Mozilla Foundation's browser is technically better than IE in many fields, including web standards compliance, speed, security, features. . . It is also highly customisable with a wealth of add-ons available on the Internet. While the default browser got too comfortable with its monopoly, Firefox rapidly and constantly implemented new features, including tabbed browsing[3], a search field. . .

[1] Most technical terms are defined in the glossary.

[2] With a market share of 35% for Firefox 3, 11% for IE 6, 34% for IE 7 and 2.3% for IE 8 [Paul, 2009].

[3] Which was actually invented by Opera.

Safari (8%) saw a dramatic increase in its market share, thanks to iPhone’s success. While it used to be available only on the Mac Os X platform, a PC version of Safari was released in 2007. Safari’s differentiation factors are similar to Apple’s key success factors and favours speed and simplicity. Its fourth version (in beta as time of writing) features a minimalist design and is the fastest browser on Mac, only a bit slower than Chrome on Windows [Absous, 2009]. Safari also provides very good standard compliance and was one of the first browser to fully comply with the Acid2 benchmark, in April 2005[4].

Opera (1%), yet having a very small user base, is in some technical fields better than its rivals. It is by instance considered the most standard-compliant browser[5], and its 9.5 version (released in June 2008) was the fastest on Windows. Opera is also very present on mobile devices, through Opera Mobile and Opera Mini.

Date	IE	Firefox	Safari	Chrome	Opera
April 2007	78.3%	15.5%	4.6%	*NA*	0.4%
April 2008	74.8%	17.8%	5.8%	*NA*	0.7%
March 2009	66.8%	22.1%	8.2%	1.2%	0.7%

Table A.1: Browser market share in the USA by date (from NetApplications)

II. Chrome: “a fresh take on the browser”[6]

On September 1, 2008, Google announced a new browser, Chrome [Pichai and Upson, 2008a], followed the next day by the first public beta release for Microsoft Windows (XP and late only) in 43 languages.

This surprising move was accompanied by a comic book drawn by Scott McCloud [Cloud, 2008] which explained the reason why the firm released this product. *The reader is strongly encouraged to read this freely available*[7] *comics*, which will give a better understanding of some technical terms as well as the novelty of Chrome. Google’s browser has introduced many *incremental innovations* improving software’s stability, speed and user experience.

Speed is undoubtedly the most important innovation in Chrome. As Google reckons, today’s web is made up of RIA (rich internet applications). These complex web applications are developed in JavaScript, a programming language. While other browsers have focused their effort on implemented new features and improving overall speed[8], Google chose to specifically focus on JavaScript speed. A dedicated team leaded by Lars Bak developed *from scratch* a new JavaScript engine named V8. This component includes many optimisations: efficient garbage collection, dynamic code generation and “hidden classes”. . . which have made Chrome by far the fastest browser at the time of its release[9].

Googlers also improved the browser’s **stability**, through another component change: the sandboxing mechanism. On other browser, all tabs share the same memory: if only one of these pages crashes, the whole program will crash, with all other opened tabs. In Chrome, a dedicated *sandbox* is created for each web page, and JavaScript applications are executed inside [Cloud, 2008, p.3-9]. A tab can only affect the sandbox in which it runs, i.e. if it crashes, the user will still be able to use the browser. This behaviour is very similar to an OS’. The sandboxing mechanism also greatly improves security [Dang, 2009].

[4] Safari 4 already scores 100/100 on the Acid3 benchmark [Lilly, 2009].

[5] It was the first browser to fully implement the CSS standards.

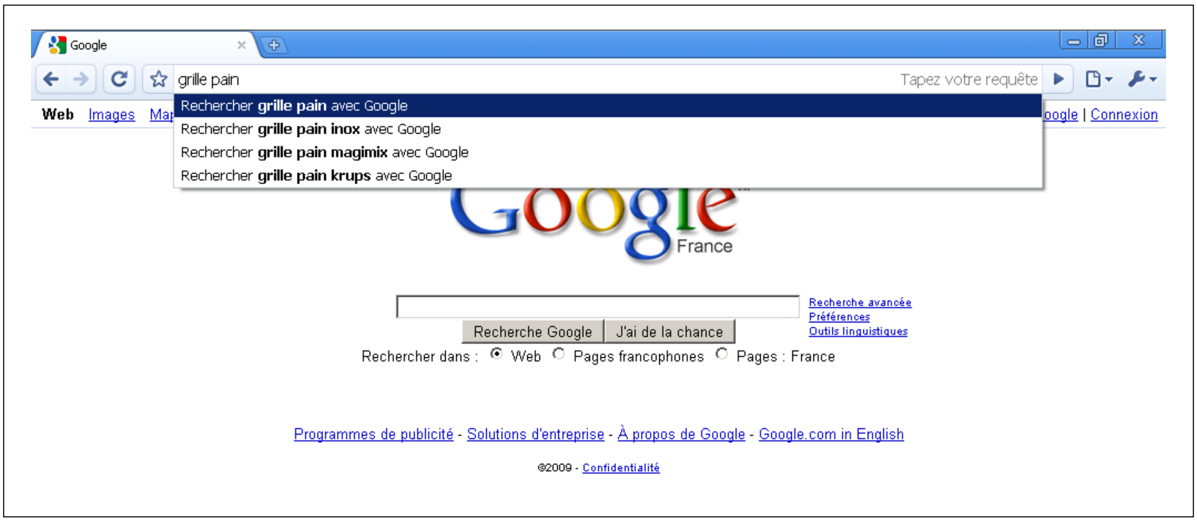
[6] Pichai and Upson [2008a]

[7] `http://www.google.com/googlebooks/chrome/`

[8] Layout engine operations (Webkit in Chrome and Safari, Gecko in Firefox, Trident in Internet Explorer and Presto in Opera) are very time-consuming, which explains why browsers have always tried to optimise them.

[9] In the beginning of March 2009, V8 is still 40 times faster than IE 7, and 4 times faster than IE 8 beta 2 [Lilly, 2009].

Chrome's **user experience** makes it also appears like an operating system. With no toolbars and no status bar, its User Interface (UI) philosophy glorifies efficiency, like Google Search's homepage. As said in the comic book: "if [the user] can just *ignore* the browser, we've done a good job" [Cloud, 2008, p. 24] Chrome also includes an intelligent address bar, very similar to Firefox's. When the user types information (for instance the word "toaster"), requests are automatically sent to Google Suggest[10] and return popular sites and related queries to help the user (e.g., an already visited toaster store, or a Google search with keywords including "toaster store").



grille pain

Google
Tapez votre requête
Rechercher **grille pain** avec Google
Rechercher **grille pain inox** avec Google
Rechercher **grille pain magimix** avec Google
Rechercher **grille pain krups** avec Google
Web
Images
Map
oogle | Connexion
Google
France
Recherche avancée
Préférences
Outils linguistiques
Recherche Google
J'ai de la chance
Rechercher dans : ● Web ○ Pages francophones ○ Pages : France
Programmes de publicité - Solutions d'entreprise - À propos de Google - Google.com in English
©2009 - Confidentialité

Figure A.1: Google Chrome showing the Suggest feature. Displayed suggestions offer direct links to search requests.

III. Chrome, a free and open source application

Whereas Internet Explorer, Opera and Safari are *proprietary* software, Chrome and Firefox are *free software*. Understanding the consequences of this difference is vital: we will explain what Free Software (FS) is with a two-steps approach:

1. What is a program source code? Why does it matter?
2. What are Free and Open Source Software (FOSS)[11]?

We will then briefly discuss the implications of having released Chrome as FOSS.

1. Explaining FOSS

1.1. Source code: a human-readable version of software

To understand why does source code matter, we will compare computer programs to cakes[12] (freely adapted from Scott Colford Colford [2008]):

[10] The same kind of feature already exists on Google's homepage, where suggestions are made when the user begins to type characters in the search field.

[11] It is beyond the scope of this paper to comprehensively discuss all aspects of FOSS. To get more information, please refer to Wikipedia.

[12] Other allegories include the "software as a car" version [Options, 2009].

> Imagine that you have to buy a cake for a friend's birthday. You go to the bakery to see what they've got on display and you find a lovely white cake with a beautiful yellow icing with "Happy Birthday!" flowing across it. The cake is absolutely perfect, although you would better have a green icing. There are other pre-made cakes available but they have even more things wrong with them. You can order a custom cake, of course, but it's more expensive, takes a week and you doubt that the bakery can meet your specific demands.
>
> What will you do? Why don't you bake your own cake? It would be pretty hard if you had to guess at the ingredients and just experiment with various ratios of flour, butter and sugar. Sure would it be great to get the list of ingredients, as well as the *recipe*. . . But will the baker accept to give this trade secret?

If the cake is our computer program, then the recipe is its *source code*. Without its source code, it is almost impossible to modify or understand an application: the unidirectional process that makes source code (recipe) become a program (cake) is called the *compilation* (cooking).

1.2. FS and OSS

Free software. FS has been defined by Richard Stallman in February 1986. A program is free if users are given:

Freedom 0. The freedom to **run** the program, for any purpose.

Freedom 1. The freedom to **study** how the program works, and adapt it to your needs.

Freedom 2. The freedom to **redistribute** copies so you can help your neighbor.

Freedom 3. The freedom to **improve** the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits.

As explained before, freedoms 1 and 3 require source code to be available because studying and modifying software without its source code is highly impractical. The basic idea behind the Free Software Foundation (FSF) philosophy is freedom, which refers to ethical or moral values.

OSS. The OSI (Open Source Initiative) movement promotes a more practical and business-friendly philosophy [OSI, 2007]:

> Open source is a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in.

FOSS. Besides these axiological differences, both philosophies are in fact very similar. In this paper, we will use the expression *Free and Open Source Software (FOSS)* to refer to the following concept [Wikipedia, 2009]:

> Free and open source software is software which is liberally licensed to grant the right of users to study, change, and improve its design through the availability of its source code.

FOSS are very often available *gratis*. Yet as Richard Stallman puts its,"Free software is a matter of liberty, not price. To understand the concept, you should think of 'free' as in *free speech*, not as in *free beer*". Cost-free (*gratis*) software are generally labelled "freeware", whereas software *libre* are called "free software"[13].

The simplest way to make a program free software is to put it in the public domain, uncopyrighted. Yet doing so does not deal with many other problems (notably authors will want credit for creating the source code, and will offer no express or implied warranty — since differences exist between the

[13] In French, "free" does not convey this ambiguity: free software are referred as "*logiciel libre*".

countries regarding the legal value of “public domain”), consequently free software are usually released under a FOSS license. The most popular licenses are: the GNU/GPL license, the BSD license, the MPL (Mozilla Public License), the Apache license, the Creative Commons license (CC-BY and CC-BY-SA) and the MIT license.

Key FOSS include GNU/Linux, OpenOffice.org, Apache, Wordpress, Drupal, Firefox, MySQL, The GIMP, and LaTeX, with which this paper is written.

2. The consequences for Google

While all browsers are free as in beer[14], only Firefox and Chrome are FOSS.

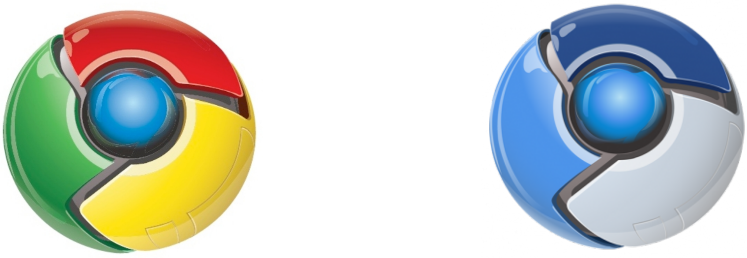


Figure A.2: Chrome (left) and Chromium (right) logo

Strictly speaking, Chrome is a proprietary software. It is in fact based on another Google application, Chromium. The actual free software is Chromium, but since Chrome only adds the Google logo and some minor UI controls to Chromium [McAllister, 2008], we will conveniently refer to Chrome as being FOSS.

Google has released Chromium’s source code under a free license (the BSD license as printed in Appendix A). It means that everyone can download the program’s source code and potentially distribute its own version of Chrome. This also means that the sandboxing feature, the V8 JavaScript engine. . . can be reused, even in proprietary software like Internet Explorer or Safari. Googlers may have worked for other companies like Microsoft or Apple. . . While Google could have patented the sandboxing and some of V8’s enhancements, or even kept them as trade secrets, it has chosen to release them. More than that, it has given everybody the *right* to use them in any other programs.

These points are summarised in Figure A.3.

[14] It has always been the case for all browsers but Opera which was made free on September 20, 2005 [Opera, 2005].

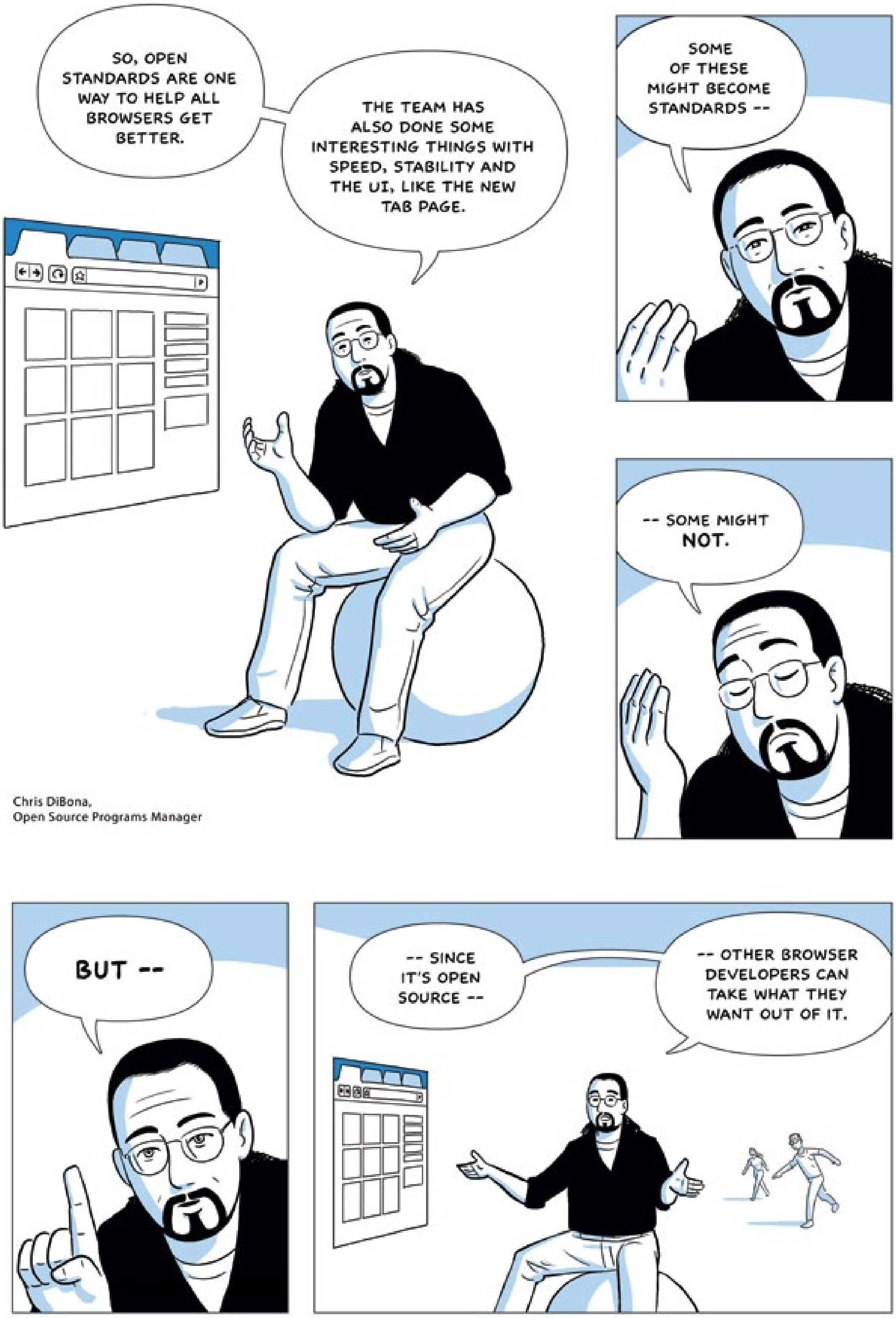

SO, OPEN STANDARDS ARE ONE WAY TO HELP ALL BROWSERS GET BETTER.
THE TEAM HAS ALSO DONE SOME INTERESTING THINGS WITH SPEED, STABILITY AND THE UI, LIKE THE NEW TAB PAGE.
Chris DiBona,
Open Source Programs Manager
SOME OF THESE MIGHT BECOME STANDARDS --
-- SOME MIGHT NOT.
BUT --
-- SINCE IT'S OPEN SOURCE --
-- OTHER BROWSER DEVELOPERS CAN TAKE WHAT THEY WANT OUT OF IT.

Figure A.3: Extract from the Chrome comics explaining that it is a free software (words by the Google Chrome team, comics adaptation by Scott McCloud).

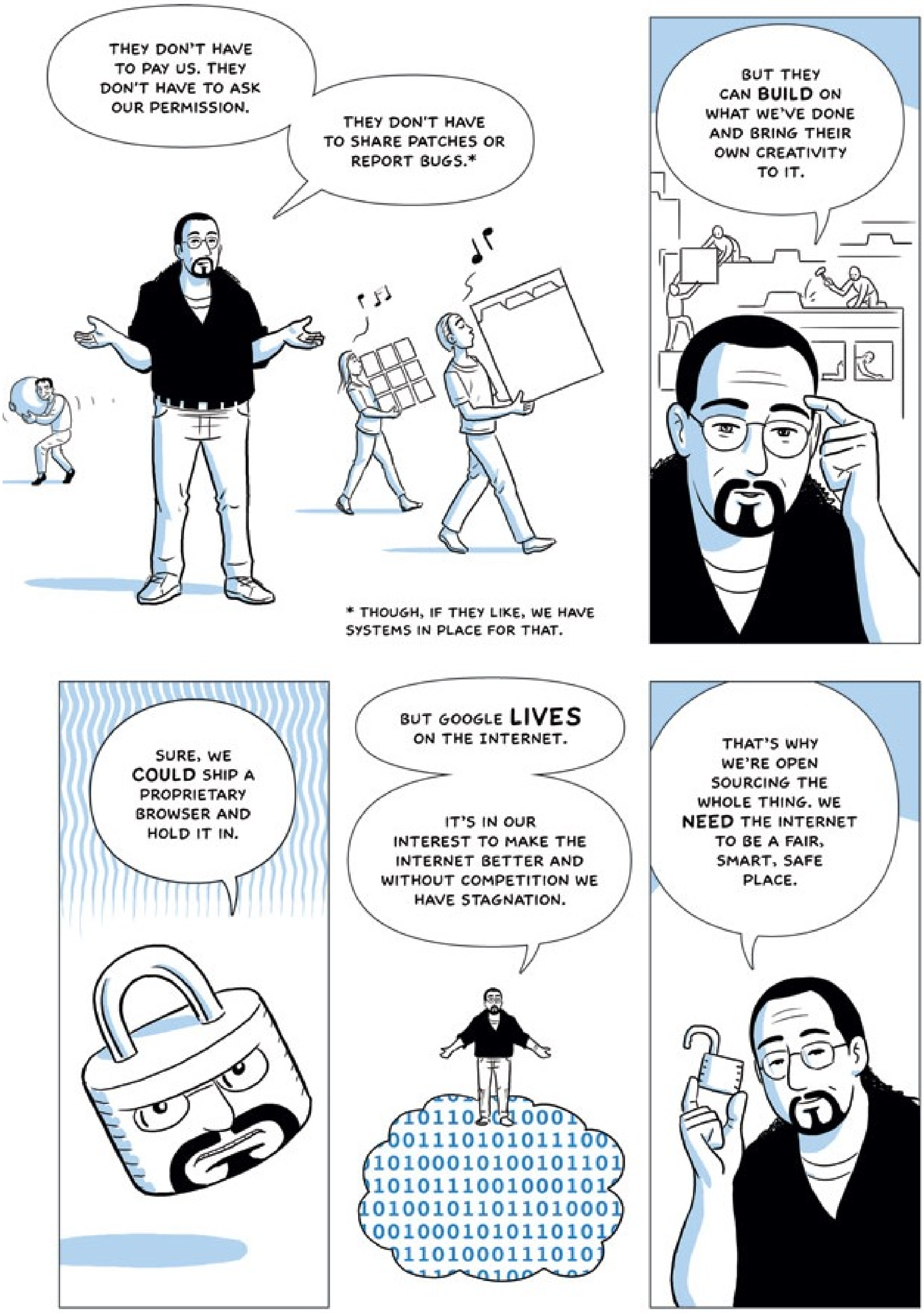

THEY DON'T HAVE TO PAY US. THEY DON'T HAVE TO ASK OUR PERMISSION.
THEY DON'T HAVE TO SHARE PATCHES OR REPORT BUGS.*
* THOUGH, IF THEY LIKE, WE HAVE SYSTEMS IN PLACE FOR THAT.
BUT THEY CAN BUILD ON WHAT WE'VE DONE AND BRING THEIR OWN CREATIVITY TO IT.
SURE, WE COULD SHIP A PROPRIETARY BROWSER AND HOLD IT IN.
BUT GOOGLE LIVES ON THE INTERNET.
IT'S IN OUR INTEREST TO MAKE THE INTERNET BETTER AND WITHOUT COMPETITION WE HAVE STAGNATION.
THAT'S WHY WE'RE OPEN SOURCING THE WHOLE THING. WE NEED THE INTERNET TO BE A FAIR, SMART, SAFE PLACE.

Figure A.3: Continued

B. Case resolution: Google Chrome as a launching platform

The Google Chrome case shows a strategic gap, which will be featured through a syllogism:

- **Major premise:** Google is a company: its goal is to generate sustainable profit.
- **Minor premise:** Chrome is FOSS, which means that Google's employees have potentially worked for anybody, since the program's improvements are not protected and are also available free of charge, a fact that will not help Google diversify its cash generating activities (97% of its revenues comes from advertising).
- **Conclusion:** Releasing Chrome as FOSS must reveal something about Google's diversification strategy.

This dissertation is to address the following problem: **why has Google chosen to release Chrome as free software?**

I. The analysis of facts and symptoms

The subject discussed by this dissertation being very narrow, it is wise to use Occam's razor and understand two main topics, where many facts and symptoms can be consolidated to give a wider view of the company's global strategy: why is Google often using the FOSS development model (§1.)? What is Chrome's raison d'être in the firm's strategy (§2.)?

We will then analyse the decision to make Chrome FOSS using a variation of Michael Porter' five forces model (§3.). The *dominant design* model will finally explain the links between Google's choice, openness, and innovation in the Internet sector (§4.).

1. Free software at Google

1.1. An anamnesis: openness in the history of Internet

FOSS used by Google. Developing or using open source applications is nothing new at Google. On the contrary, the company is doing an *anamnesis*: it is in particular thanks to FOSS like GNU/Linux (a free OS) and Apache (a web server) that the firm has been able to built itself[1]. Without a free OS like GNU/Linux, the firm would have never been able to create such a scalable architecture.

Open standards. Moreover, since Google is a web company, using the competency tree analysis shows that openness is in the firm's roots. Navigating the Internet is possible only because standardized and open formats exist (XHTML, CSS, PDF, PNG. . .). It has not always been the case: during the first browser war, Internet Explorer and Netscape were promoting their own standards, making the work

[1] For a comprehensive and very interesting history of Google, see Vise and Malseed [2006]

of web developers more difficult. Later on, the W3C (World Wide Web Consortium)'s was created to "develop interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential". However, since adhering to standards is not compulsory, many websites are still not cross-browser compatible: during all the time it was dominating the market, IE did not improve its standards compliance.

Open standards and data gathering. Google's willingness to foster open standards is logical. Its search engine was not able to reference Flash web applications before 2008[2] [Adler and Stipins, 2008]. It is also possible to use Flash along with other format to obfuscate data, which makes referencing considerably harder, or even impossible. It is thus clear that Google wants to promote open standards to facilitate the gathering of data.

Open source and open standards, two loosely linked notions. The notion of open source and open standards are not very closely linked: Opera, a proprietary software, is characterised by very good standard compliance. Yet theoretically, FOSS will rather use open standards, since in such application, there is no incentives to use an abstruse standards. Internet has been build upon free software: all the main network protocol have been implemented in FOSS: DNS, FTP, HTTP... What is more, the Internet have remained interoperable thanks to public intervention, particularly against Microsoft's monopoly strategy; as Janet Abbatte says (cited in [Rayport and Heyward, 2009]):

> Critical to the success of the Internet was the fact that these government agencies worked to ensure an open, interoperable Internet, despite the subsequent efforts, many years later, of companies like Microsoft, AOL, and WorldCom, which attempted to break up the Internet into separate, proprietary networks.

The relationships between FOSS and open standards is indeed very complex. This paper will rely on the description provided in Hoe [2006, p. 5], which explains that "FOSS is useful for popularising open standards":

> FOSS can play a useful role in popularizing an open standard. A FOSS implementation of a standard usually results in an open and free-working reference implementation. A lot of the benefits of open standards are negated, if its only implementation is a closed and proprietary one. The availability of a FOSS implementation will spur quicker adoption and acceptance of the standard as everyone has easy access to the implementation of the standard and so can try and test it out. A very good example of this is the Internet HTTP standard. One reason why this service became universally accepted is that very early on there were free and open implementations of both the HTTP server (e.g., National Center for Supercomputing Applications or NCSA HTTPd, Apache) and client (e.g., NCSA Mosaic).

1.2. The practical advantages of FOSS for Google

Google's reputation. Even if Google is never referring to the *free software* movement, *open source* naturally refer to positively framed values: freedom, openness, diffusion of knowledge... They inevitably tone down its monopoly status, especially vis-à-vis its arch-rival, Microsoft, which is in comparison seen as the "great Satan" [Foundation, 2009], for many reasons that include: bundling of new computers with the Windows OS, non-compliance of many Microsoft products with standards, law suits against companies developing FOSS, and absence of collaboration between Microsoft and the FOSS movement[3].

Creating a Google community. It is difficult to perfectly understand the implications of using a FOSS development model as regards HR and adoption. Yet the socio-technical approach [Akrich, 1991] may provide a good understanding of this subject.

[2] Restrictions still exist, images are notably not referenced. What is more, it is nearly impossible to make direct links to specific pages in a Flash website (which makes a search engine useless).

[3] Large companies like Intel, Sun and Novel are heavily supporting FOSS development.

Focus on the user. Google has been able to build a community, like many other companies in the high-tech sector. Having released many software as open source, the firm encourages the most competent users to modify its products (or propose improvements or report bugs), whilst most users will just use them. By doing this, Google is placing the passionate user in the centre of its production process. They will in turn become the firm's *spokespersons* or evangelists[4]. What is more, instead of resorting to a "classic" lead user approach, Google has been able to outsource many of its services to the community, which in turn improve the user's satisfaction.

Focus on the developer. Promoting FOSS also makes it easier to recruit competent developers. For example, the Google Summer of Code is a "global program that offers student developers stipends to write code for various open source software projects". FOSS helps Google make talent-scouting [Ippolita, 2008]. Supporting the FOSS development model encourages external developers to contribute to other projects (e.g. Android), especially when they are specialised in such a field. Since its software can be exhaustively understood, developers will more frequently use Google's Application programming interfaces (APIs), by instance to develop applications on the Android platform. This can be described as a strategy of *collaboration-exploitation* [Ippolita, 2008], as Google is subsequently integrating these developers.

Free Software also offers an extrinsic valorisation of Googlers' work [Girard, 2008]: the gratification a developer gains is the satisfaction of having done a good work, and his peer's admiration: to be praised, a solution must be understood and admired; only open and commented source code can offer such gratification.

2. The raison d'être of Chrome

2.1. Encouraging innovation in the Internet sector

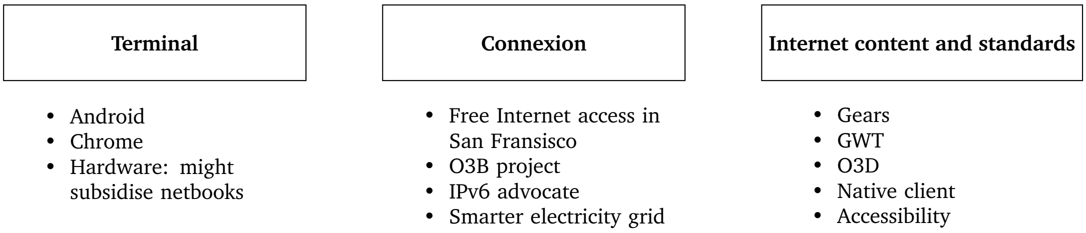
In 2004, Google publicly said it will not release its own Google-branded browser, counting on other browsers to innovate [Olsen and Dumout, 2004]. It seems that Google found that change was too slow, since only four years later, Chrome was launched.

Vertical integration. Google may be tightly controlling the Internet searching market (it represents 67.5% of the US market in March 2009, Yahoo 20.5% and Microsoft 8.3% [Chitu, 2009b]); it still depends on what Philipp Lenssen calls the *web information distribution chain*. In the same way industries depends on steady input of raw materials, Google's contingency lies in every person's access to the Internet (see the risk factors in the annual report of Google [2007, p. 27-28]). Microsoft could for instance introduce feature hurting Google's products, Firefox could remove its search engine from the default settings. As Eric Schmidt reckons, there is a "defensive component" in this launch, to stop Microsoft from "balkanizing" the web [Freeland, 2008]. "The more parts of the chain Google owns, the more control it can exert, and potentially, the less it is risked to be attacked or pressured by enemies along the way" explains Philipp Lenssen. This is the reason why it is fuelling innovation in the Internet sector, to improve the reliability of the whole value chain [Minto, 2009]:

> The sophistication of web applications is increasing much faster than the corresponding power of the browsers. It's like having lots of high-performance sports cars using rutted, bumpy roads. And yet no car manufacturer has ever invested in road building.

The Mountain View firm thus resorts to a common scheme in the industrial economy: the *vertical integration*. In this case though, integration does not mean tight control, but willingness to foster innovation and to foster an open ecosystem (it is exactly the same strategy with Android in the mobile sector [faberNovel, 2008, p. 23]. As it is *living* on the Internet [Cloud, 2008, p. 37], making it grow is obviously in its interest. Like an industry, the firm prefer to develop in-house products and to have a direct influence on the market. As Jérémy Milhau says, Chrome carries a "friendly" message: "adapt or vanish!": other browser will have to innovate to stay in the course.

[4] For instance, Gmail was launched through a system of invitation.



Terminal

- Android
- Chrome
- Hardware: might subsidise netbooks

Connexion

- Free Internet access in San Fransisco
- O3B project
- IPv6 advocate
- Smarter electricity grid

Internet content and standards

- Gears
- GWT
- O3D
- Native client
- Accessibility

Figure B.1: A simplified approximation of the web information value chain shows that Google is higly investing in all layers.

Terminal. Google's willingness to foster innovation is present on the terminal:

- **Hardware:** Google might subsidise the purchase of Linux netbooks [Metz, 2009].
- **Software:** it has already build its own operating system, Android, another free software, and released Chrome.

Connexion to the Internet. The firm is publicly advocating for incremental innovation in the Internet sector. To consolidate data about this fact, the OSI model (Open Systems Interconnection Reference Model) gives a good approximation of areas where Google is acting:

- **Physical (layer 1):** Google is offering free Internet access in San Francisco [Mills, 2007], and has invested in the O3B network, a project aimed at providing Internet satellite coverage to "the other three billion" people in areas like Africa and the Middle East [Rhoads, 2008].
  The firm is also promoting "open standards for a smart energy grid", applauding the $4.5 billion funding by the Congress to build a smarter electricity grid that can empower consumers with information about their electricity consumption [Wingo, 2009].
- **Network (layer 3):** Google's services are already accessible through the Internet Protocol version 6 (IPv6) [Marsan, 2009]. In a post on its Public Policy Blog, the firm advocates for the use of IPv6, mainly because with the present version of the protocol, the Internet is projected to run out of IP addresses in 2011. Temporary workarounds exists but as Google says they "undermine the Internet's open architecture and 'innovation without permission' ethos" [Erik Kline and Derek Slater, 2009].

What is more, Dr. Vinton G. Cerf. Vint, co-designer of the TCP/IP protocol, is Google's Chief Internet Evangelist [Müller, 2009]. The company is also contributing research papers on many subjects to the World Wide Web Conference, in particular about accessibility [Spector, 2009].

2.2. Strengthening Google's ecosystem

Advertising represent 97% of the Google's revenues. Its business model is based on a win-win relationship with customers: *(a)* it exchanges free services (Gmail, Blogspot, Google Map...) *(b)* for audience (eyeballs) and consumer data (behavioural marketing).

Consumer marketing. Chrome implements an intelligent bar, very much like the one included in Firefox. As the user types data in the address bar, a list of suggestions is printed (search requests, or direct link to websites). Whereas Firefox computes this list from offline data, every keystroke is sent to Google, along with the IP address and any Google account information stored in cookies. This is indeed a very clever way to efficiently gather behavioural data (especially since Chrome has only one input area for entering both URLs and search queries), which may then be used for *behavioural targeting*, a very efficient marketing technique which considerably increases the return on investment of advertising campaigns. This feature, along with the “usage statistics and crash reporting” feature, has raised privacy concerns even if it can be disabled in Chrome’s preferences [Conley, 2008].

This ability to outsource a job to the crowd has been named *crowdsourcing* [Howe, 2006] and is deeply rooted in Google’s history.

Audience. According to a study by AT Internet Institute, the Google toolbar accounted for 20.1% of the search engine’s visits [AT Internet Institute, 2009]. The firm’s strategy in this field is to have browsers or original equipment manufacturers (OEMs) pre-includes its search engine in their products. For instance, Apple Safari, Opera Mobile [Opera, 2008] and Firefox use Google by default[5], which consequently increases the search engine’s audience.

Comparing ROIs. Since Google already had many agreements, one should consider why it has chosen to develop Chrome internally. After all, Firefox was already part of Google’s ecosystem: an agreement was settled to make the search engine the default choice. It resulted in royalties of $80 million [Mozilla Foundation, 2007], accounting for about 91% of Mozilla Foundation’s revenues and only 0.5% of Google’s expenses.

This is the reason why it is interesting to compare both investments with approximated figures[6]: Table B.1 shows that Chrome is a heavy project that must thus serve a more ambitious purpose, which will be described thereafter.

	Firefox	Chrome
Market share in the US - March 2009	22.1%	1.2%
Cost in 2008	$80 m	approx. $15 m

Table B.1: A comparison of the Chrome and Firefox investment shows that Google must have ambitious objectives. Data are very rough approximations.

2.3. Google envisions a cloud computing paradigm

Understanding the cloud. Another key notion to this paper is the *cloud computing*[7].

> What technologists like to call ’the cloud’ is the idea of computing on demand. Just as you turn on a faucet to get water or plug into a wall to get electricity, the features and functions of stand-alone computers today can be streamed to you over the Internet. [...] In the cloud, applications are accessible anywhere, anytime, and storage becomes, for all intents and purposes, infinite. [...]
>
> Cloud computing represents a new way to deploy computing technology to give users the ability to access, work on, share, and store information using the Internet. The cloud itself is a network of data centers — each composed of many thousands of computers working together — that can perform the functions of software on a personal or business computer by providing users access to powerful applications, platforms, and services delivered over the Internet.

This definition used in a paper commissioned by Google [Rayport and Heyward, 2009] explains how the Internet is shifting towards the cloud computing paradigm.

Google in the cloud. Google’s strategy can be interpreted as visionary. The firm has deliberately embraced the cloud computing paradigm, even if most details of this vision are emergent and unplanned. Many signs show that Google is strongly vesting its power, with a twofold tactic: *(a)* simplifying the user’s access to the Internet, *(b)* fuelling innovation in the key components of cloud computing.

[5] Safari 4 also implements the Google Suggest feature.

[6] Google is said to have dedicated 100 software engineers to the Chrome project. According to Glassdoor.com, the average salary for a Google engineer is about $100,000. We then add other related costs, arbitrary fixed at about 50% of the projet.

[7] To better understand this concept, the reader should refer to Appendix C for a more complete definition by Google.

Access. Google's goal in this area is to transform the OS into an *abstraction layer*[8], and to place the browser in the centre. The terminal would be limited to a minimalist OS (which could resemble Android, developed by Google) with a browser (Chrome). Chrome and Android are thereby building up a Google-powered cloud computing terminal [Sadun, 2009].

Components. Google is not only fuelling innovation in the global Internet, it is also providing its own solutions to cloud computing-related issues as a *web shaper*. Nothing is really linking these innovations, apart from the fact they are all FOSS improving some aspects of existing technologies:

- **Gears** (licensed under the BSD license) adds new features to browsers[9], including the local storage of data (offline capabilities) and the ability to run JavaScript in the background to improve its performance (speed).
- **V8** (BSD license) is a new JavaScript engine implemented in Chrome. When launched, it was significantly faster than JScript (the JavaScript engine implemented in Internet Explorer), SpiderMonkey (Firefox), and JavaScriptCore (Safari).
- **Native Client** (BSD license) is a research project for running x86 native code in web applications. Instead of being constrained by the browser, web applications would be able to use directly and most importantly securely the CPU. This dramatic speed boost offers new horizon to make more complex computations, like 3D image manipulation. Google has not given many information about this project, and it remains unclear if it is aimed at competing with already existing RIA solutions. Yet the Native Client (NaCl) project could result in making the OS layer irrelevant [Jaber, 2008; Braux, 2008].
- **GWT** (Apache 2.0 license) allows developers "to quickly build and maintain complex yet highly performant JavaScript front-end applications in the Java programming language."
- **O3D** (BSD license) was released in April 2009. It is an open source browser plug-in expressely aimed at "encouraging the discussion within the graphics and web communities about a new open web standard on 3D graphics for the web" [Bridge and Tavares, 2009].

The goal of this paper is not to discuss all these products launched by Google. Yet they show that the firm has chosen to address specific problems step by step. There is no doubt that all these products have in common Google's vision, which is, according to Linus Upson, a Google Engineering director: "there are things you can do in desktop apps that you can't do in Web apps. We're working very hard to close that gap, so anything you can do in a desktop application you can do safely and securely from a Web application." [Shankland, 2009b].

3. Understanding Chrome as FOSS: the six forces analysis

In §1., we have seen that FOSS had many practical advantages for Google. In §2., we presented the coherence of Chrome in Google's general strategy. The six forces model, an extension to Michael Porter's five forces analysis, will now provide a very good understanding of Google's strategic objectives for Chrome as FOSS.

3.1. Suppliers: efficiency in the development process

FOSS used in Chrome. Chrome has not been built from scratch. It reuses many existing components, most importantly WebKit (a layout engine) and has about 25 dependencies, with different licenses (presented in Appendix B).

[8] From Wikipedia: "An abstraction layer (or abstraction level) is a way of hiding the implementation details of a particular set of functionality."

[9] It is pre-included in Chrome and can be installed in IE, Safari and Firefox.

A free and numerous labour force. Suppliers are *developers*, since software are made up of algorithms. As explained in §1.2., the FOSS development model involves many practical advantages: the existence of a Chrome community will make Google benefit from free support, free application coding, free debugging, free advertising. . . done by passionate users. With an open source development method, the Mountain View company has secured a free and numerous labour force, and outsourced a part of its r&d department: other developers improving Chrome's dependencies are indirectly working for companies using their work (Apple, Google in the case of WebKit). Using already existing components has considerably quickened and reduced the cost of Chrome's development, thus enabling Googlers to focus on their most important innovations.

Attracting developers from other browsers. As explained in §1.2., Google's FOSS strategy is also a mean to attract developers. A great majority of Googlers working on Chrome were Firefox contributors [Kennedy, 2008]. There have been fierce debates about whether or not Firefox is the most innovative browser[10]. Google could integrate specialised developers from other browser. What is more, the firm is also giving more power to developers, because it gives them the possibility to test new features on Chrome (as in Firefox). As a *learning organisation*, the company can now better understand web standards and build a network model of innovation: this knowledge will enhance its ability to control the market.

3.2. Substitute products: increasing market penetration

The rise of netbooks and smartphones. Rather than finding substitute to the browser, something rather theoretical because it has indeed no substitute (dedicated clients for email, RSS or music will never offer a complete alternative, especially in a cloud computing paradigm), it is more interesting to analyse the desktop market. Chrome was first launched only on the Windows platform, which is logical since it is still the dominant OS. Yet the dramatic increase in market share of *netbooks* and smartphones explains why Google has not turn a blind eye on other platforms, where resources are more constrained. Developers have to implement lighter version of their browsers: Fennec (Firefox for Mobile), Safari Mobile, Opera Mobile, Opera Mini, Internet Explorer Mobile, Android browser, which is also based on WebKit. Google has even developed its own image renderer for constrained mobile devices, Skia [Yin, 2008]. It is a brand new vector graphics renderer, released as free software in the same time as Chromium.

FOSS: an efficient way to increase portability. FOSS are more easily developed on many platforms, as Google CEO Eric Schmidt publicly said [Barak, 2009]. For instance, the Linux kernel[11] runs on about one hundred architectures. This practical advantage over proprietary software have many causes: *lead users* can easily adapt the application to make it run on their hardware, since the source code is available.

The threat of netbooks or smartphones does not exist for Google: manufacturers could even build their own Chrome version, which will increase its market adoption. This is not only theoretical: Android has already been ported to netbooks (while it was designed to run on specific smartphones), because of its open source status [Sadun, 2009].

3.3. Rivalry among existing competitors: defying Microsoft

A hit aimed at Microsoft. Before Chrome, Google had never fully used advertising, mainly because its faith lied in one of its principles: "focus on the user, and all else will follow". It was the first product in the firm's history to be heavily supported by an advertising campaign: starting April 2009, IE users (only them) are specifically targeted on Google's homepage, to make them install Chrome [Laporte, 2009]. The firm has even launched a TV advertising campaign [Barnett, 2009], consequently reckoning that Chrome have failed — for now — to provide serious threat, as shown in Table B.2. Based on this very aggressive marketing campaign, it is obvious that ambitious market share objectives has been set.

[10] Among the main criticisms, Firefox's memory usage is said to be worst than Chrome's, and Gecko slower than WebKit [Pierce, 2009].

[11] The kernel is the core of an operating system.

Date	Net Applications (USA)	AT Internet Institute (Europe)
November 2008	0.83%	0.9%
December 2008	1.04%	1.1%
January 2009	1.12%	1.2%
February 2009	1.15%	1.3%
March 2009	1.23%	1.4%
April 2009	1.42%	1.7%

Table B.2: Chrome market share by date : a slow but steady adoption rate

This strong move is transformed by FOSS, which enables Chrome to appear as a positive move for the user, toning down Google's increasing dominance. Through this development model, the Mountain View company will also be able to gather all the forces against Microsoft, "coopeting" (cooperating and competing) with the other browsers. Googlers are for instance closely cooperating with WebKit developers. On the contrary of Microsoft, which has always favoured locking in customers (e.g., by bundling Windows and IE), Google's business model is based on collaboration (and exploitation): it makes it appear as kind and selfless.

Limited sunk costs. The firm has also been able to limit its sunk costs, because even if Chrome is not able to secure a consistent user base, its innovation are freely reusable. Google's objective, which to foster innovation on the Internet, would have been attained.

3.4. Users: increasing their bargaining power

Chrome will give information. In the browser market, users do not have enough information: Linus Upson, engineering director of the Chrome project, said that "the biggest challenge all [browsers] face is that most people don't even know what a browser is or that there's choice" [Gilbertson, 2008]. IE is still the default browser on most computers; Firefox has been adopted by innovators, early adopters and part of the early majority. These actors are not sufficient for securing Chrome's success.

What is more, the bargaining power of customers is very limited, because of *path dependence*: the bundling of Microsoft Windows and Internet Explorer makes it difficult for unskilled Internet user to understand why they should change their default browser to another one. Many websites appear differently on non-IE browser, because of its non-compliance with web standards. This is the reason why the gist of the matter is Microsoft's ability to lock the market. As Google's CEO Eric Schmidt said [Metz, 2009]:

> The problem has to do with Microsoft's ability to use its Windows monopoly to restrict a user's fair choice. . . Anything that Microsoft would do that would eliminate user choice with respect to the way search engines and internet browsers are distributed — for which it was previously found guilty — would be of concern.

Reducing the switching costs from IE. Google's goal is to increase information about browsers, so that users will stop using IE. Even if it is really trivial to download Chrome or Firefox, Google said that is was planning to settle agreements with OEMs in order to include Chrome on newly built PCs, which will boost Chrome's market adoption. Google's tactic is simple: reducing the switching costs will increase the competition in the web browser sector. The firm hopes that the user will make the good choice, i.e. use a more innovative and more standard-compliant browser. Making Chrome a free software can also increase its ability to bargain with OEMs, since they would be able to easily implement their own customisation: we could for instance imagine an HP-branded browser based on Chromium.

3.5. Entry of new competitors: increasing the competition

Entry of new competitors has long been inhibited by the shipping of IE with Windows. Why dedicate resource to a browser while Microsoft is predominant?

Yet the European Commission formally objected to the bundling and put Microsoft on notice in January 2009. Launched by Opera's complaint [Opera, 2009b], the proceedings have seen Mozilla joining in as "third-party"[Kawamoto, 2009], entitling it to see confidential documents in the case, and to voice objections. Google has also joined the antitrust case [Helft, 2009b], which could result in the unbundling of IE and Windows. What is sure is that the chosen solution(s) would lower entry costs, for instance by giving access to the OS distribution channel.

Since Chrome is FOSS, anyone could develop its own version of browser, with very few resources. Google is not only encouraging new entrants, it is also helping them. This would subsequently increase the competition in the browser market.

3.6. Public policies: building up a positive frame

Whereas public policies may not be obvious in the browser market, Google seems to have taken the helm of being the public actor, lobbying and working for the common good to increase the competition. In a post by Google's European Policy Manager, the firm offers "an analogy to the French Revolutionary goals of *liberté, égalité and fraternité*, saying openness, open standards, and open source must be the three guiding principles for the Net to achieve this revolutionary promise." [Müller, 2009].

Releasing Chrome as FOSS has been a wise move, as it is politically far easier to support the FS philosophy than a proprietary one: from the State's point of view, Google's use of FOSS positively frames its actions. Since Chromium is free, a State may even publicly endorse a modified or localised version.

Even if Chrome includes features hampering the user's privacy[12] (something that most users will not care about), it is possible to develop another browser based on Chromium. For instance, SRWare Iron[13] is a Chromium-based browser without any feature raising privacy concerns. FOSS consequently improves Google's transparency: if Chrome would have been proprietary, data protection specialists would have raised questions about the protection of privacy. The firm has already used FOSS to reassure customers: Google Update have been open sourced in April 2008 to address concerns over its functioning and "to make [its] purpose totally transparent" [Chitu, 2009c].

3.7. Conclusion: Google increases the forces to foster competition

As we have seen in the previous sections, FOSS in Chrome will have a large impact on the browser industry structure. The combined effects of this strategic choice are summarised in Figure B.2, which shows that it will ultimately increase the competition in the sector. FOSS is used by Google in order to efficiently develop a strong concurrent to Microsoft's IE. This method of development will ensure Chrome's success. By increasing the power of customers, suppliers, substitutes and potential new entrants on the one hand, it has increased the competition between companies in the browser industry on the other.

Google has already succeeded in a way, because this increased competition has had one consequence: browsers have very quickly imitated some of Chrome's new features (Firefox will be multiprocess according to Cabello [2009], and could even take Chromium's networking stack), notably in the field of speed:

- Safari 4 will embark a faster JavaScript engine, Nitro [Absous, 2009].
- Opera 10 promises a 30 percent performance boost [Lilly, 2009].
- Firefox 3.5 also includes a new and faster JavaScript engine, TraceMonkey.

[12] However, on September 6 the German Federal Office for Information Security warned internet users about Chrome [Lenssen, 2008], advising that it should not be used because of its beta status (removed on December 11 [Pichai and Upson, 2008b]) and that Google's move raised competition-related concerns.

[13] `http://www.srware.net/en/software_srware_iron.php`

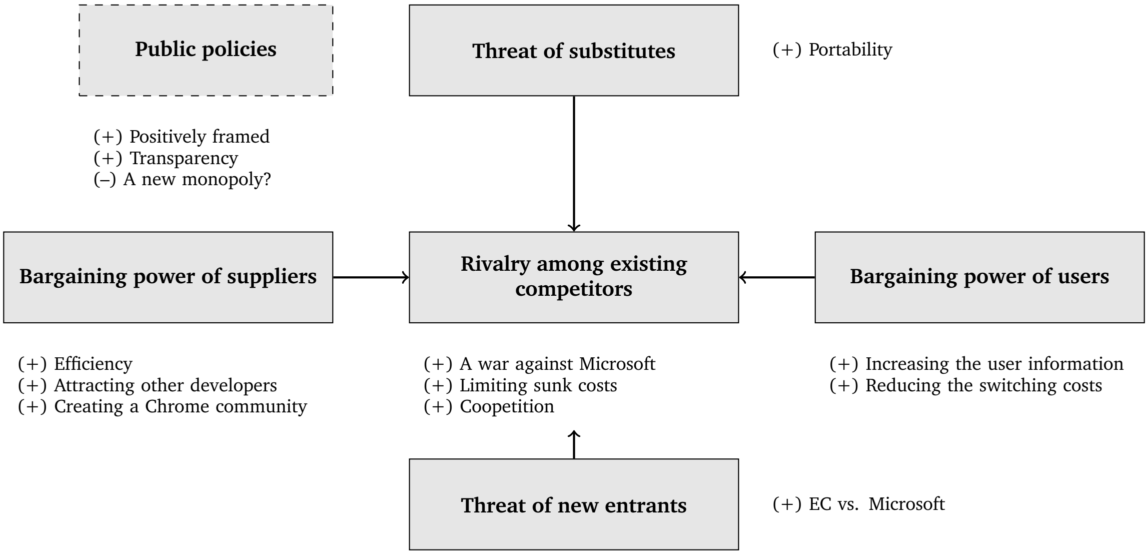

Public policies
(+) Positively framed
(+) Transparency
(–) A new monopoly?
Threat of substitutes
(+) Portability
Bargaining power of suppliers
(+) Efficiency
(+) Attracting other developers
(+) Creating a Chrome community
Rivalry among existing competitors
(+) A war against Microsoft
(+) Limiting sunk costs
(+) Coopetition
Bargaining power of users
(+) Increasing the user information
(+) Reducing the switching costs
Threat of new entrants
(+) EC vs. Microsoft

Figure B.2: FOSS in Chrome influences industry structure by increasing competition

4. The dominant design model: understanding the links between competition, openness and innovation in the Internet sector

By making Chrome FOSS, Google is willing to change the existing *dominant design*, which is not prepared for the cloud computing paradigm. The value of a dominant design is composed of its stand-alone technological utility (SAV), its installed base (IB) and the availability of complementary goods (CG).

$$Value = SAV + IB + CG$$

IE used to be the worst browser in terms of standards compliance. It is easy to understand how this has undermined innovation on the Internet. Webdevelopers have to create sites for the largest possible audience. As explained in §1.1., they were obliged to use complicated workarounds to make their products appear identically on all the browser[14].

It exists an *IE dominant design*, which Google wants to replace with an alternative, more innovative dominant design. This design already exists, and is supported by innovative browsers (Firefox, Opera, Safari), yet its overall value is smaller than Microsoft's one. As a result, Chrome's goal is to increase the value of each of its components, especially its IB and its CG.

4.1. Stand-alone technological value: new standards for more powerful web applications

New features and better portability. Because of the lack of innovation in the Internet sectors, most websites does not use the new features provided by new open standards. These improvements includes the ability to make complex manipulations on videos, with open standards like the video tag element in HTML5 [Nitot, 2009].

Google explained how these specifications paved the way for mobile web programs (cloud computing) and helped it develop "fantastic new applications", including the ability to use web applications while offline: "your users can enjoy fast, capable web apps that they can access from any device, without the need to copy their data from place to place or worry about installing software or being online." [Nicolaou, 2009]. Google strongly promotes HTML5 on its blog, giving advice on how to use

[14] It is still the case today, notably for some designing techniques implemented by the CSS (Cascading Style Sheets) specifications, that Internet Explorer still does not support. The page `http://quirksmode.org/compatibility.html`, maintained by Peter-Paul Koch and not surprisingly sponsored by Google, lists all issues relative to browser adoption of web standards [Quirksmode, 2009].

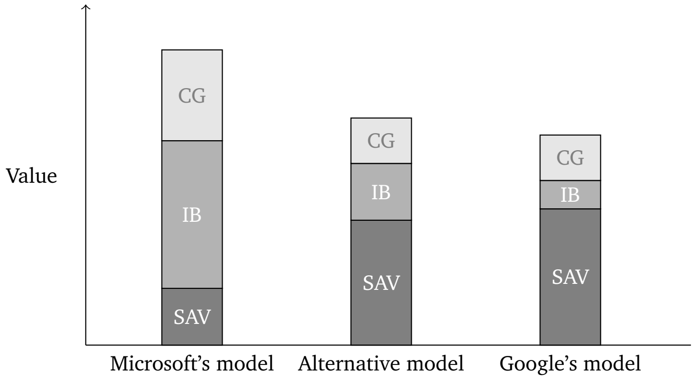


Figure B.3: Comparing the IE dominant design and the Google-supported innovative design shows that Chrome have to increase its user base (IB) and its complementary goods (CG) to diffuse its standalone technological value (SAV). Google's model and the already existing innovative model (Firefox, Opera, Safari) are complementary and will join forces to provide better browsers, yet for convenience they are separately presented. Values are arbitrary.

it [Grieve, 2009]. It hosted a technical talk entitled "The Open Web Goes Mobile" made by Peter-Paul Koch, which dealt with browser compatibility on mobile devices [Souders, 2009], which Google reckons as being "even worse than it is on the desktop".

This shows how standards, browsers, web applications and cloud computing are tightly interconnected: *browsers must support new open standards to enhance the development of more powerful web applications, portable on every devices*.

New formats: 3D and SVG. To take a concrete example, IE still[15] does not support the SVG standards, whilst all many other browsers implement this format. Since IE is the dominant design, there is no reason why developers should use SVG. This is evidently slowing the web [Baker, 2009b]: for example, this standard enables the creation of complex image animations and has enabled Google to offer a drawing feature in Google Docs [Glenning, 2009]. According to Serge Cheminade, the SVG format could really be the future of Internet, as it enables web developers to build animations, fonts, images... using an open format.

Google is becoming a key player in the 3D animation sectors. It has for instance created a website to show how powerful V8 is, especially when compared with its counterparts[16]. The website `www.chromeexperiments.com` is a showcase of Chrome's ability to make very complex computations, including fluid 3D rendering [McNamee, 2009]. Google also released in April 2009 the O3D plug-in, presented page 19.

4.2. Increasing the user base to foster the diffusion of innovation

A launching platform. FOSS is a launching platform used to ensure the diffusion of innovation: since it is easy for another developer to get the source code of Chrome or one of Google's plug-in (Gears, O3D), anybody can implement them in new applications[17], which consequently *increase the diffusion rate* of these technologies, and their user base. The firm is able to implement some of its technologies directly into Chrome[18], in order to fuel their diffusion.

[15] The user can install a plugin to handle SVG files, but it is unprobable that unskilled users will do that.

[16] It interestingly said in the announcing post, that "although you don't need Google Chrome to view the site, some of the experiments may run slower, or incorrectly, in *older* browsers" (I emphasise) [Koblin, 2009].

[17] Even in web applications competing with Google's: Zoho is using Gears, and MySpace announced that it would offer its member offline access using the same Google technology [Berlind, 2008].

[18] Gears is no more a dedicated project, it is part of Chrome's source code.

Creative destruction. *Creative destruction* of value is a very common process in Google's business model: while other software companies (Microsoft and Adobe for instance) use proprietary standards (Silverlight and Flash), the Internet giant opens up the market and destroys the value of existing products by releasing its own, free solution. As Serge Cheminade explains, developing programs based on a proprietary technology like Flash involves two risks:

1. A financing risk: in most case, Software development kit (SDK) or assimilated are sold.
2. A technologic risk: the standard owner controls the innovation.

This is the reason why open standards like those set by the W3C are theoretically more innovative. Proprietary standards are more difficult to diffuse: to use Flash, the user must install a plug-in. The raison d'être of FOSS at Google is to eliminate all those risk and to freely give innovation to the market. Table B.3 shows that these proprietary standards are pervasive. Making true Opera's Vision Statement's wishful thinking, "*standards are much like the spirit of the Internet; a structure on which human innovation can prosper to the benefit of everyone.*" [Opera, 2009a] will not be an easy task.

	December 2008
Adobe Flash	99%
Java	81%
Microsoft Windows Media	70%
Apple QuickTime	62%
Adobe Shockwave	55%
Real One	39%
SVG	*6%*

Table B.3: Market penetration of players by Internet-enabled PCs (from Incorporated [2008]. These figures are somewhat biased because they do not encompass all devices, notably the iPhone.)

Fighting Microsoft's dominant design. Using 3D on the Internet before required the use of proprietary and licensed software like Flash or Silverlight. IE does not support SVG, but implement another Microsoft-backed standard, VML. Since Microsoft's business model is based on the lock-in of a majority stake in the market[19], we understand why Google tries to push the SVG format by releasing its own browser as FOSS to increase the installed base of alternative browser. What is more, path dependency explains why corporate user are reluctant to change their browser: e.g., according to Forrester Research, many web applications still only work with IE [Dignan, 2009].

The lack of innovation (*stand-alone technological value*) is a serious threat to Google's interest. This is the reason why Google pushes its pawn in every format war related to the Internet: since it reckons that SAV does not suffice to gain market adoption, FOSS is enrolled to rapidly increase the user base, which will, as the firm hopes, create a new dominant design where innovation would have a greater place.

4.3. Complementary goods: the browser as a development tool

The browser, a developing tool. Another key point in the *dominant design* model is the importance of *complementary goods*. The browser can now be considered a development tool (or an IDE): most browsers includes developer features to help them create efficient RIAs. The richness of Chrome and Google's new APIs will allow new interactions between Internet and the terminal, "all of this could usher in a new wave of more sophisticated web applications, cheaper and more dynamic to use" [Hartmann, 2009]. What is more, Chrome must be linked to Android. They are designed to be the next development tools, since in a cloud computing paradigm, all the applications will be accessed through the internet.

[19] For example, the use of closed format `.doc` document is slowing the adoption of other products like OpenOffice.org.

"Google is paving the way for developers to build browser-based applications that can run on any mobile platform, as opposed to having to build separate versions of their applications in order to support those same mobile platforms" [Berlind, 2008]: it is thus using FOSS to increase the diffusion of this model, notably against the Apple model (iPhone applications only works on Apple's mobile).

Attracting *prosumers*. Webdevelopers are obviously the key actors in this field, as they develop applications executed from a browser. Google's goal is to attract these "*prosumers*" (producers and consumers) on its platform. By releasing Chrome as an open source software, and soon including the possibility to develop extensions, Google is evidently willing to create a developing community around its software (we already explained how this could be possible in §3.1.), as well as integrating third-party developers. Attracting developers on its platform would create a wealth of new applications specifically designed for more innovative browsers, which would consequently increase the number of complementary goods, thus increasing its value.

II. Strategic analysis

The previous section analysed and interpreted many facts and symptoms regarding Google's strategy as regards FOSS in Chrome. This section will now present a summarised strategic analysis, followed by a strategic diagnosis.

1. FOSS shows how Google is protecting itself

Google's true innovations are kept secret. A very interesting point about innovation at Google is made in *Le modèle Google : une révolution du management* [Girard, 2008, p. 129]: innovation is not always profitable. According to Peter Drucker, only true innovations are not copyable. This means that releasing Chrome's source code is unimportant to Google, because its inherent value is very small.

Indeed, Google's true innovations are kept hidden. They include *scalability*: *the firm's strongest competitive advantage is its ability to efficiently manage large farms of servers to gather enormous amount of information*: "Google is designed to scale well to extremely large data sets." [Brin and Page, 1998]. Most innovations have never been released, nor patented: GWS (Google Web Server) is one of most important components in Google's infrastructure[20], and one of its most guarded secrets even if it often releases anecdotic information [Dingman, 2008]. Some of these innovations have been patented though, including a book scanning process [Schonfeld, 2009].

Applying the *operational innovation* concept (as explained by Hammer [2005]) to our case shows that Google's true assets is its ability to innovate (by using FOSS). This is the reason why it does not matter if small incremental innovation like Chrome or V8 are released into the market. Releasing them as FOSS keeps from asking how Google has been able to build this monopoly. Indeed, this concept offers a comprehensive understanding of Google's strategy: whereas in old industries (including the OS market), it is possible to use a monopoly to introduce proprietary (closed) standards which will then secure income; since the Internet favours inter-operability, openness and transmission of information, the only way to create so-called standards is to have them open. *It only implies that, in the Internet era, standards must be open. They could then generate income for a monopoly.*

Openness will make the Internet grow. Dr. Vinton G. Cerf. Vint explained that "the openness of the network is going to be the engine to create new wealth." [Müller, 2009]. As we showed in §I., FOSS in Chrome will:*(a)* increase the diffusion of innovation, *(b)* increase the audience of Googe's services, *(c)* increase the quality of web applications, *(d)* help Google secure the web information value chain, *(e)* ensure that nobody is able to control the value chain. As a consequence, the Internet will obviously grow.

[20] Google's servers represented about 10.6% of the active sites, Apache 46.2% and Microsoft's IIS 35.6% in October 2008 [Survey, 2008].

Google creates value because of its ability to privatise the Internet. According to Michael Porter it is only "the uses of the Internet that ultimately create economic value" [Porter, 2001, P. 65]: consequently, Google is using its position to create value. As Google itself reckons in its 2006 analyst day presentation (which notes were mistakenly released) [Google, 2006, slide 35], its revenues are based on a fairly simple equation:

$$Revenues = Users \times \frac{Queries}{User} \times \frac{Ads}{Queries} \times \frac{Clicks}{Ads} \times \frac{Revenue}{Click}$$

Which can be simplified in:

$$Revenues = Users \times \frac{Revenue}{User} = Users \times Toll$$

Advertising revenues. This shows that since Google has been able to privatise the Internet, making the Internet grow is in its interest, because it can be considered a *toll*[21]. More possibilities gives more services, more services gives tighter competition, which will make Internet services more attractible. This will in turn make advertising revenues grow.

Data gathering. As Serge Cheminade puts it, the more an Internet user is connected, the more Google is able to gather information[22] (including behavioural data, as explained in §) and sell ads. This will ameliorate its search engine results too.

2. FOSS shows how Google will find new sources of monetisation

Google release its innovation as FOSS because it increases their diffusion (see §4.), which helps them become standards. According to Michael Porter, "much of the economic value created by market places derives from the standards they establish. [...] But once these standards are put in place, the added value of the market place may be limited" [Porter, 2001, p. 70]. Indeed, a competitor's ability to shape the market have always been a decisive advantage.

However, Google knows that its revenues are not diversified and reckons in its annual report [Google, 2007, p. 25] that "new technologies could block [its] ads"[23]. Google must found other source of revenues.

Licensing web-applications. Another source of revenues for Google would be the licensing of web-applications. In a cloud computing era, Google's unique competitive advantage would be for instance its ability to include a wealth of services in a unified environment (it is already showing signs of this, see for instance Shankland [2009a]). Given its brand weight, it is very probable that the company would very easily sell SaaS (Software as a service). By doing so, the firm would go from maintaining *operational effectiveness* to *strategic positioning*. Indeed, there are huge opportunities; according to Gartner Inc., a global information technology research and advisory firm, cloud computing will achieve $56 billion in 2009, and $150 billion in 2013. One should not wonder why Google is willing to take a stake of this threefold increase.

3. Diagnosis: Chrome as FOSS, a *launching platform*

Based on this strategic analysis, we are able to make the following diagnostic about Chrome as FOSS:

[21] This notion is taken from faberNovel [2008, p. 38], which develops an adamantine demonstration about how Google has been able to privatise the Internet.

[22] Some of the new technologies Google is pushing will also make this task easier.: the firm is said to test a new interface for its Google Search, which will use the AJAX technology [Chitu, 2009a].

[23] AdBlock, an extension for Firefox, blocks the display of all ads.

- It protects Google's ecosystem.
- It prepares Google's offensive on the cloud computing market, which will diversify the firm's revenues.

Chrome is FOSS because it will not generate value by itself. Its purpose is to serve as a *launching platform* for innovation in the Internet. These innovations will indirectly generate cash-flows.

C. Conclusion: limits and further research

I. Scenarios for Chrome's future

It is difficult to draw scenarios for an application's future, especially for Chrome, since many threats could arise:

- Will Microsoft remains idle vis-à-vis this threat on its border?
- Will the European Commission rule that Microsoft should not bundle IE and Windows?
- What will be the answer of Safari, Firefox and Opera to Chrome?
- Will Google manage to settle agreements with OEMs? For what cost?
- Open standards depends on their adoption

The two main reasons why it is difficult to determine scenarios are on the one hand Microsoft's unpredictable reaction, and on the other Google's market position in a cloud computing paradigm?

Google vs. Microsoft. Google has chosen to gently compete with its arch-rival and have relatively spared three of Microsoft's core businesses: Windows, the Office suite and the server software have not seen a direct strike from the Mountain View firm. From this point of view, Google is adopting a blitzkrieg tactic: *(a)* it concentrates its forces on the online market (it has not properly entered the offline market), *(b)* dedicates relatively small teams and small resources to project like Google Documents[1], *(c)* and remains very mobile because of its multiple activities on the Internet.

However Google's future obviously lies in its battle against Microsoft. As Pierre Fremaux told me, one should not be mislead by the proportion of advertising revenues in Google's income statement. Even if they have only decreased from 99% to 97% between 2007 and 2008 [Google, 2008], the fact that Google has been able to generate $667 million from its licensing activities should be considered a clear strike on Microsoft's main businesses. This is clearly related to the cloud computing paradigm, even if Microsoft is also investing in this area: it plans to release a web-based version of its Office suite [Fried, 2008].

Android has already been experimented on a netbook and a PC. Many OEMs have already announced that they will sell netbooks with Android[2]: in this market too, Google prefers not to frontally confront Microsoft, but takes aggressive indirect step by increasing its market penetration in the netbook market [Hickins, 2009], and trying to conquer other geographical area, including China [Ducourtieux, 2009].

Google, a new monopoly?. We have seen in §2. that Google was placing its pawns (including Chrome) for a cloud computing era. Instead of directly competing with Microsoft, it is concentrating its forces in order to remain the most important actor on the Internet. Yet since cloud computing is still burgeoning, new actors (Amazon by instance) could also profit from it. Will Google be able to sell licenses and consequently diversify its activities?

There are three scenarios regarding Chrome's future, from Google's perspective:

[1] According to Google [2006, slide 33], these projects represent less than 30% of Google's resources.

[2] Hewlett-Packard is notably said to "study" Android [Vaughan-Nichols, 2009], Asus is rumoured to contemplate the idea [de La Grandière, 2009].

1. **Baseline scenario**: Chrome represents 10% market share by 2011, and Google is able to spread innovation in the market. This is, according to me, the most probable scenario.
2. **Optimistic scenario**: Chrome and Chromium-based browsers controls 30% of the market by 2011. Google holds a monopoly in the cloud computing paradigm. One could reasonably argue that such a situation would not be positive, neither for Google nor for consumers.
3. **Pessimistic scenario**: Chrome is seen as a too aggressive step. An alliance between many Internet actors hampers Google's businesses.

1. Baseline scenario: a strengthened position

Market share. Chrome's success cannot be measured with its sole market share. Its goal is to foster innovation (and standards adoption) in the Internet market, which means a smaller market share for IE, and a greater one for more innovative browsers. There are little threats to Google's offensive on the browser market. The true question is how many time it will take Microsoft to efficiently answer to this threat. Google Chrome would have taken a 10% market share in 2011 thanks to many OEM agreements[3].

It is easy to understand that many customers will choose Chrome over Firefox, Safari or Opera because they naturally associate Google and the Internet. The European Commission would have ruled that Microsoft should not bundle its browser IE (Internet Explorer) with Windows[4], leaving an enormous new market for Google Chrome and Firefox. As Mitchell Baker said, "One of the results of the Windows / IE integration is that millions of people believe that the 'blue e' icon IS the Internet" [Baker, 2009a]: Google would have a greater advantage because its brand is commonly associated with the Internet.

Internet growth. The renewed competition would have helped Google spread new innovations. JavaScript engines would run far quicker, which would have considerably helped the development of web applications. Through the development of new Internet technologies, the traffic would have considerably expanded, strengthening the firm's predominance. It would have secured a very impressive user base for its search engine, especially since the renewed search agreement with Firefox (35% market share), which would have continuously drowned Firefox users to Google Search.

Cloud computing. Even if the cloud computing paradigm is still far, most specialists reckon that it will completely overhaul the current Internet market. With innovative and open browsers representing about 50% of the market, it is difficult to see any special threat to Google's diversifying of revenues.

2. Optimistic scenario: a new Google monopoly

This scenario is only optimistic from Google's perspective.

Market share. A truly optimistic scenario would see an overwhelming success of Chrome. Google's browser would represent one third of the market in 2011, through agreements with all the major OEMs, and public endorsement from officials. Localised and specific version, based on Chromium, would represent another third of the market, with for instance a significant part of developers from Firefox leaving Mozilla to build a free browser called "Aluminium"[5].

[3] It has already been able to secure such agreements for its toolbar, for example with Packard Bell [Aguila, 2006].

[4] It has already said that the integration of IE does violate EU law [Baker, 2009b]

[5] An extension called *Chromifox* already exists and replicates Chrome's user interface: `https://addons.mozilla.org/en-US/firefox/addon/8782`

Internet growth. Another key move for Google would be to fully integrate a version of Android and Chrome[6]. Google's innovations in all the major layers of the web information chain would have strengthened its tight control over the Internet. It could engage in more anti-competitive behaviour given its ability to privatise the Internet, which should be considered a common good.

Cloud computing. In a cloud computing environment, controlling the two-thirds of the browser market would make Google the new Microsoft. The firm could even drop some of aspects of its "openness" rhetoric, to introduce a sort of lock-in.

3. Pessimistic scenario: an overambitious step revealing monopolistic behaviour

Chrome already succeeded. In my opinion, Google has *already* fulfilled many of its goals: releasing Chrome subsequently fostered the innovation in the browser market, as explained page 22. It is very probable that Chrome will continue to gather users (even if its market adoption could have been more impressive). As said earlier, the main question is Microsoft's answer, yet it seems highly improbably that the Redmond giant will be able to keep users from switching from IE.

Over-ambition. According to some reported comments from competitors, Google could have gone too far with this launch. It has already to deal with the turmoil with newspapers (whose work is syndicated on Google News), editors (whose books are to be integrated in the Google Library). Privacy concerns are far from being addressed: Gmail, Google Documents, Google Search, Google Sites, Google Health. . . are gathering massive amount of data. This could considerably hamper Chrome's development. When looking at Google's strategy, could this aggressive entry into the browser market considered as overambitious? The firm's discourse indeed seems often overconfident, and can be resented as arrogant by its competitors.

Google's Achilles' heel. As a consequence, a pessimistic scenario in Google's perspective would be, paradoxically, a complete success of Google, with Chrome massively drowning users from IE. Google would become the most important actor in the cloud computing paradigm.

Options exists, yet highly improbable to me: a complete destruction of Microsoft's monopoly, followed by the released of Windows as FOSS[7]. We could also imagine an alliance between many actors to build a new search engine[8].

These are just speculations, but in a pessimistic scenario an antitrust case would be launched against Google[9]. Even if Google fosters openness in its speeches, it remains that the mass gathering of information raises many concerns. It is interesting to note that for now the protection of data kept by Google have not been criticised[10]. The smallest affair in this field would have very strong repercussions for the firm's position in the Internet sector.

II. Limits of this work

Despite all my efforts (being in Dublin, where its European Headquarters are located, contacting other contributor companies), I have not been able to directly talk with a Googler. One should not forget that strategy remains secret and hidden, while in the same time source codes can be released transparently. I did not get any financial information too, and could not approximate ROI.

I have limited technical background, solely based on my own researched, I may have misinterpreted some technical notions. Cloud computing is a very recent concept. Google's strategy is obviously aiming

[6] Sergey Brin has already announced a Chrome-like version for Android [Shankland, 2008a].

[7] Articles have already contemplated the idea [Babcock, 2009].

[8] Google's exclusive license rights on the patent hold by Stanford University are not eternal.

[9] Authors already explained how it could be possible [Clemons, 2009], like the one launched in April 2009 regarding the Google Book deal [Helft, 2009a].

[10] Despite a privacy error discovered in May 2009 that inadvertently shared some documents [Kincaid, 2009].

towards this, yet as some specialists reckons, many problems have not yet been fixed (most importantly the issue of permanent connexion to the Internet) and innovation always carries uncertainty. As a consequence, this paper could be too confident in the future of this new paradigm.

Further subjects. Given the limited size of this paper, some particular aspects may have been too quickly studied, or models could have been wrongly understood. Other models could have been used, including Brian Arthur's model of increasing returns. Some studies cited by this paper could also be biased or be based on an incorrect methodology.

The most important limit regarding this paper is that I could not take into accounts the difficulties Google will be faced with. Many subjects have also not been studied in depth by this paper, including:

- The network model of innovation at Google
- Financial aspects of the cloud computing paradigm
- The true links between openness, open standards and open source
- The links between Google, Mozilla and Apple
- Google as a new Microsoft ?
- The diversification of Google's revenues

Bibliography

Vincent Absous. Safari 4 fonce sur le JavaScript. *Mac Génération*, March 3 2009.

Ron Adler and Janis Stipins. Google learns to crawl Flash. *The Official Google Blog*, June 2008.

Nicolas Aguila. Google partenaire de Packard-Bell. *Tom's Guide*, November 2006.

Madelein Akrich. L'analyse socio-technique. In D. Vinck, editor, *La gestion de la recherche*, pages 339–353. De Boeck, 1991.

AT Internet Institute. Février 2009 Orange sous le seuil des 1% en part de visites. *Baromètre des moteurs*, February 2009.

Charles Babcock. Why Windows Must Go Open Source. *Information Week*, January 2009.

Mitchell Baker. EC Principle 5: Microsoft must educate people about other browsers. *Lizard Wrangling*, April 2009a.

Mitchell Baker. "IE must comply with web standards." (Opera has suggested that Microsoft must support web standards they have promised to support). *Lizard Wrangling*, April 2009b.

Sylvie Barak. 2009 is an Android year declares Google's Schmidt. *The Inquirer*, April 2009.

Emma Barnett. Google launches first TV ads. *Telegraph*, May 2009.

David Berlind. Could There Be More To Google, Android, Chrome, & Gears Than Meets The Eye? *Information Week*, September 2008.

Bertrand Braux. Google dévoile une alternative à Flash, Silverlight et Java. *01net Pro*, December 2008.

Henry Bridge and Gregg Tavares. Guest post: 3D graphics in the browser. *The Chromium Blog*, April 2009.

Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Master's thesis, Stanford University, 1998.

Percy Cabello. Future Firefox will be multiprocess. *Mozilla Links*, May 2009.

Frédéric Cavazza. Y aura-t-il un avant et un après Google Chrome ? *FredCavazza.net*, September 2008.

Alex Chitu. Google Search Prepares for Switching to Ajax. *Google System*, April 2009a.

Alex Chitu. Google dominates search more than ever. *Google Watch*, April 2009b.

Alex Chitu. Google Update, Open Sourced. *Google System*, April 2009c.

Eric Clemons. What An Antitrust Case Against Google Might Look Like. *TechCrunch*, March 2009.

Scott Mc Cloud. *Google Chrome*. Google, September 2008.

Scot Colford. Explaining Free and Open Source Software. *asis&t Bulletin*, December 2008.

Chris Conley. Paying for Chrome with Personal Data. *ACLU of Nothern California - Issues*, October 2008.

Alan Dang. Exclusive Interview: Google Chrome's Chromium Core Explored. *Tom's hardware*, April 2009.

Arnauld de La Grandière. Android s'émancipe du téléphone. *Mac Génération*, April 2009.

Larry Dignan. Browser wars could improve productivity, but the enterprise still loves IE6. *ZDNet.com*, April 2009.

Jonathan Dingman. Interview with Bharat Mediratta About the Google Web Server. *Google News, Inside*, May 2008.

Cécile Ducourtieux. Avec le lancement de Windows 7, Microsoft joue gros. *Le Monde*, May 2009.

IPv6 Software Geek Erik Kline and Policy Analyst Derek Slater. Why the next generation Internet Protocol matters. *Google Public Policy Blog*, March 18 2009.

faberNovel. *Google's key success factors*. faberNovel, November 2008.

Free Software Foundation. Is Microsoft the Great Satan?, January 2009. URL `http://www.gnu.org/philosophy/microsoft.html`.

Chrystia Freeland. Google admits Chrome designed to repel Microsoft in browser wars. *Financial Times (FT.com)*, September 2008.

Ina Fried. Web-based Office to work on Macs, iPhone. *CNET News*, November 2008.

Scott Gilbertson. Google Chrome Out of Beta, Official 1.0 Release Available. *Wired.com*, December 2008.

Bernard Girard. *Le Modèle Google : une Révolution du Management*. M2 Éditions, 2nd edition, April 2008.

Tony Glenning. Drawing on your creativity in Docs. *The Google Docs Blog*, March 2009.

Google. Google 2006 Analyst day presentation. In *Analyst day presentation*. Google, 2006.

Google. Google Annual Report 2007, 2007.

Google. Google Annual Report 2007, 2008.

Andrew Grieve. Gmail for mobile HTML5 Series: Using AppCache to Launch Offline - Part 1. *Google Code Blog*, April 2009.

Michael Hammer. Six Steps to Operational Innovation. *Harvard Management Update*, 10(4), April 2005.

Matthäus Krzykowski & Daniel Hartmann. Android netbooks on their way, likely by 2010. *VentureBeat*, January 2009.

Miguel Helft. Justice Dept. Opens Antitrust Inquiry Into Google Books Deal. *The New York Times*, April 2009a.

Miguel Helft. Google Joins Europe Case Against Microsoft. *The New York Times*, February 2009b.

Michael Hickins. Note to Microsoft: Your Netbook Days Are Numbered. *Google Watch*, April 2009.

Nah Soo Hoe. *Free/Open Source Software - Open Standards*. United Nations Development Programme – Asia-Pacific Development - ELSEVIER, 2006. ISBN ISBN-13: 978-81-312-0538-9.

Jeff Howe. The Rise of Crowdsourcing. *Wired.com*, June 2006.

Adobe Systems Incorporated. Flash Player Penetration, December 2008. URL `http://www.adobe.com/products/player_census/flashplayer`.

Ippolita. *La face cachée de Google*. Manuels Payot. Payot et Rivages, March 2008.

Sami Jaber. Avec Native Client, Google invente l'OS dans le navigateur. *DNG Consulting*, December 2008.

Dawn Kawamoto. Mozilla to weigh in on EU's Microsoft case. *CNET News*, February 2009.

Niall Kennedy. The story behind Google Chrome. *Niall Kennedy's Weblog*, September 2008.

Jason Kincaid. Google Privacy Blunder Shares Your Docs Without Permission. *TechCrunch*, March 2009.

Aaron Koblin. Chrome Experiments are here. *Official Google Blog*, March 18 2009.

Christophe Laporte. Google veut faire switcher les utilisateurs d'Internet Explorer. *Mac Génération*, April 2009.

Philipp Lenssen. *Google Chrome Receives Heavy Criticism in Germany*. Google Blogoscoped, September 2008.

Paul Lilly. Browser Battle: Nine Browsers of Today and Tomorrow Compared. *Maximum PC*, March 3 2009.

Carolyn Duffy Marsan. Google: IPv6 is easy, not expensive. *Network World*, March 25 2009.

Neil McAllister. Building Google Chrome: A first look. *Infoworld*, September 2008.

Dean McNamee. Monster Mesh - A Google Chrome Experiment. *Google Code Blog*, March 18 2009.

Cade Metz. Google boss backs subsidized Linuxbooks. *PCs & Chips*, March 2009.

Elinor Mills. EarthLink, San Francisco finalize Wi-Fi contract. *CNET News*, January 5 2007.

Rob Minto. The genius behind Google's web browser. *Financial Times (FT.com)*, March 27 2009.

Mozilla Foundation. *Independent Auditors' Report and Consolidated Financial Statements*. Mozilla Foundation and Subsidiary, 2007.

Sebastian Müller. If It Isn't Open, It Isn't The Internet. *Google European Public Policy Blog*, April 2009.

Net Applications. Browser statistics, March 2009.

Alex Nicolaou. HTML5 and WebKit pave the way for mobile web applications. *Google Code Blog*, April 2009.

Tristan Nitot. Faire de la vidéo un citoyen du Web à part entière. *Standblog*, April 2009.

Stefanie Olsen and Estelle Dumout. Pas de navigateur signé Google en perspective. *ZDNet.com*, October 2004.

Opera. Feel Free: Opera Eliminates Ad Banner and Licensing Fee. *Opera Press Releases*, September 2005.

Opera. Opera brings Google search to your pocket. *Opera Press Releases*, February 2008.

Opera. Opera's Vision, 2009a. URL `http://www.opera.com/company/vision/`.

Opera. European Commission accuses Microsoft of illegally tying Internet Explorer to Windows. *Opera Press Releases*, January 2009b.

Open Options. Software is like a car, 2009. URL `http://www.netc.org/openoptions/background/roadster.html`.

OSI. Homepage, March 2007. URL `http://www.opensource.org/`.

Ryan Paul. Firefox 3 market share crawls past IE 7 in Europe. *StatCounter*, April 2009.

Sundar Pichai and Linus Upson. A fresh take on the browser. *The Official Google Blog*, September 2008a.

Sundar Pichai and Linus Upson. Google Chrome (BETA). *The Official Google Blog*, December 2008b.

David Pierce. 6 Reasons Why Firefox Dude May Want to Switch to Chrome. *MakeUseOf.com*, April 2009.

Michael E. Porter. Strategy and the Internet. *Harvard Business Review*, March 2001.

Quirksmode. Compatibility Master Table, April 2009. URL `http://quirksmode.org/compatibility.html`.

Jeffrey F. Rayport and Andrew Heyward. Envisioning the Cloud: The Next Computing Paradigm. Technical report, Marketspace, March 2009.

Christopher Rhoads. Start-Up Seeks to Link 3 Billion to Net. *The Wall Street Journal*, September 9 2008.

Erica Sadun. Android netbook port leaves some pondering Google OS. *Ars Technica*, January 2009.

Erick Schonfeld. It Turns Out That Google Even Has A Competitive Advantage In Scanning Books. *TechCrunch*, May 2009.

Stephen Shankland. Google co-founder expects Chrome for Android. *Webware - CNET*, September 2008a.

Stephen Shankland. Chris DiBona, Google : « Chaque fois que vous utilisez Google, vous exploitez une machine qui exécute Linux ». *ZDNet.com*, June 2008b.

Stephen Shankland. Google bulks up Gmail with built-in search. *Webware - CNET*, May 2009a.

Stephen Shankland. Google plugs PC power into cloud computing. *Webware - CNET*, April 2009b.

Rajen Sheth. What we talk about when we talk about cloud computing. *Official Google Enterprise Blog*, April 2009.

Steve Souders. PPK: The Open Web Goes Mobile. *Google Code Blog*, April 2009.

Alfred Spector. 18th International World Wide Web Conference. *The Official Google Blog*, May 2009.

Web Server Survey. October 2008 Web Server Survey. Technical report, Netcraft, October 2008.

Steven J. Vaughan-Nichols. One giant step closer to the Google Linux desktop. *Computerworld Blogs*, March 31 2009.

David A. Vise and Mark Malseed. *The Google Story*. Dunod, March 09 2006.

Wikipedia. Free and open source software, April 2009. URL `http://en.wikipedia.org/wiki/Free_and_open_source_software`.

Harry Wingo. Open standards for a smart electric grid. *Google Public Policy Blog*, March 2009.

Charles Yin. Skia Source Code Released. *satine.org*, September 2008.

Glossary

For a better understanding of particular notions, the reader is strongly advised to refer to Wikipedia.

Acid2 benchmark Acid2 is a test page published and promoted by the Web Standards Project to expose web page rendering flaws in web browsers and other applications that render HTML. Named after the acid test for gold, it was developed in the spirit of Acid1, a relatively narrow test of compliance with the Cascading Style Sheets 1.0 (CSS1) standard, and was released on April 13, 2005. As with Acid1, an application passes the test if the way it displays the test page matches a reference image.

Acid3 benchmark Acid3 is a test page from the Web Standards Project that checks how well a web browser follows certain web standards, especially relating to the Document Object Model and JavaScript.

AJAX Asynchronous JavaScript and XML. Ajax is a group of interrelated web development techniques used to create interactive web applications or rich Internet applications. Ajax is said to be a client-side specification, for the creation of web pages, web sites or web applications.

API Application programming interface. An API is a set of routines, data structures, object classes and/or protocols provided by libraries and/or operating system services in order to support the building of applications.

browser war The browser wars are present and past competitions for dominance in the web browser marketplace. The term is used to denote two specific periods of time: the competition between market-dominating Netscape Navigator and its eventual defeat by Microsoft Internet Explorer during the late 1990s, and the competition from 2003 onwards between the dominating Internet Explorer and several other emerging browsers including Mozilla Firefox, Safari, Opera and, since mid-2008, Google Chrome.

BSD license This license has few restrictions compared to other free software licenses such as the GNU General Public License or even the default restrictions provided by copyright, putting it relatively closer to the public domain.

client A client is an application or system that accesses a remote service on another computer system, known as a server, by way of a network.

cloud computing Cloud computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet (see related appendix).

CPU Central processing unit

CSS Cascading Style Sheets. CSS is a style sheet language used to describe the presentation (that is, the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can be applied to any kind of XML document, including SVG and XUL.

DNS Domain Name System

Flash Adobe Flash (previously called Macromedia Flash) is a multimedia platform originally acquired by Macromedia and currently developed and distributed by Adobe Systems. Since its introduction in 1996, Flash has become a popular method for adding animation and interactivity to web pages. Flash is commonly used to create animation, advertisements, and various web page components, to integrate video into web pages, and more recently, to develop rich Internet applications.

FOSS Free and Open Source Software

FS Free Software

FSF Free Software Foundation

FTP File Transfer Protocol

GNU/GPL license The GNU General Public License (GNU GPL or simply GPL) is a widely used free software license, originally written by Richard Stallman for the GNU project. The GPL is the most popular and well-known example of the type of strong copyleft license that requires derived works to be available under the same copyleft. Under this philosophy, the GPL grants the recipients of a computer program the rights of the free software definition and uses copyleft to ensure the freedoms are preserved, even when the work is changed or added to. This is in distinction to permissive free software licenses, of which the BSD licenses are the standard examples.

Googlers Google programmers.

GWS Google Web Server. (GWS) is the name for the web server software that Google uses for their web infrastructure.

HTML5 HyperText Markup Language Version 5. HTML 5 is the fifth major revision of the core language of the World Wide Web, HTML.

HTTP Hypertext Transfer Protocol

IDE Integrated development environment. A software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of: source code editor, compiler and/or interpreter, build automation tools, and debugger

IE Internet Explorer. Microsoft's browser for desktop

IP Internet Protocol. A protocol used for communicating data across a packet-switched internetwork using the Internet Protocol Suite, also referred to as TCP/IP.

IPv6 Internet Protocol version 6

JavaScript JavaScript is a scripting language used to enable programmatic access to objects within other applications. It is primarily used in the form of client-side JavaScript for the development of dynamic websites. JavaScript is a dialect of the ECMAScript standard and is characterized as a dynamic, weakly typed, prototype-based language with first-class functions. JavaScript was influenced by many languages and was designed to look like Java, but to be easier for non-programmers to work with.

license A legal instrument governing the usage or redistribution of copyright protected software.

MPL Mozilla Public License. The MPL is characterized as a hybridization of the modified BSD license and GNU General Public License. It is the license for the Mozilla Application Suite, Mozilla Firefox, Mozilla Thunderbird and other Mozilla software.

NaCl Native Client

OEM original equipment manufacturer

OS operating system

OSI Open Source Initiative. An organization dedicated to promoting open source software.

OSI model Open Systems Interconnection Reference Model. The OSI model is an abstract description for layered communications and computer network protocol design. It was developed as part of the Open Systems Interconnection (OSI) initiative. In its most basic form, it divides network architecture into seven layers which, from top to bottom, are the Application, Presentation, Session, Transport, Network, Data-Link, and Physical Layers. It is therefore often referred to as the OSI Seven Layer Model.

PDF Portable Document File

PNG Portable Network Graphics. PNG is a bitmapped image format that employs lossless data compression. PNG was created to improve upon and replace GIF (Graphics Interchange Format) as an image-file format not requiring a patent license.

RIA rich internet applications. A good example of such applications is available at `http://280slides.com/`.

SaaS Software as a service. SaaS is a model of software deployment whereby a provider licenses an application to customers for use as a service on demand. SaaS software vendors may host the application on their own web servers or download the application to the consumer device, disabling it after use or after the on-demand contract expires. The on-demand function may be handled internally to share licenses within a firm or by a third-party application service provider (ASP) sharing licenses between firms. Examples of SaaS vendors include SAP Business ByDesign and Google Apps which provide common business applications online that are accessed from a web browser, while the software and data are stored on the servers.

SDK Software development kit

Silverlight Microsoft Silverlight is a programmable web browser plugin that enables features such as animation, vector graphics and audio-video playback that characterizes rich Internet applications.

SVG Scalable Vector Graphics. Family of specifications of text-based file format for describing two-dimensional vector graphics, both static and dynamic (interactive or animated).

TCP Transmission Control Protocol. TCP is one of the core protocols of the Internet Protocol Suite. TCP was one of the two original components, with Internet Protocol (IP), of the suite, so that the entire suite is commonly referred to as TCP/IP.

UI User Interface

VML Vector markup language. VML is an XML language used to produce vector graphics. VML was submitted as a proposed standard to the W3C in 1998 by Microsoft, Macromedia, and others.

W3C World Wide Web Consortium. The W3C is the main international standards organization for the World Wide Web (abbreviated WWW or W3). It is arranged as a consortium where member organizations maintain full-time staff for the purpose of working together in the development of standards for the World Wide Web. As of February 2008, the W3C had 434 members.

web standards Web standards is a general term for the formal standards and other technical specifications that define and describe aspects of the World Wide Web. In recent years, the term has been more frequently associated with the trend of endorsing a set of standardized best practices for building web sites, and a philosophy of web design and development that includes those methods.

XHTML Extensible Hypertext Markup Language. The XHTML, is a markup language that has the same depth of expression as HTML, but also conforms to XML syntax.

Appendix A

The modified BSD license

```
// Copyright (c) 2006-2008 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
//    * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//    * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
//    * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Appendix B

Google Chrome's dependencies and their licenses

I. Different types of licenses

There are three types of FOSS licenses:

Strong copyleft licenses. They give rights as long as the copy is also bound with the same rights. Concretely, if a company wants to use a free software under a strong copyleft license, it will have to release the source code of all its work and give the same rights (reproduction, modification, distribution).

E.g.: GNU General Public License (GNU/GPL)

Weak copyleft licenses. They are used when not all derived work has to be released under a free software license.

E.g.: the Mozilla Public License (MPL) only requires releasing the modifications made *directly* to the licensed code. Such licenses are especially useful for software libraries. To take an example, consider a library LibA released under the MPL license, and an application named SoftB using this LibA. The software can be released under a proprietary license if it used LibA without modifying anything. If it directly modified a line of LibA's source code, it must release the modification of LibA's source code under a free license, and can keep SoftB under a proprietary license.

Permissive licenses. Works under these licenses can be reproduced, adapted or distributed with very few restrictions (they are very near the public domain). They can be integrated in proprietary software.

E.g.: the BSD License (which only requires acknowledging the author), the X11 license, the Apache license

Public domain. To create free software, it is also possible to release the source code in the **public domain**, where there are no restrictions at all.

II. Chrome's dependencies

Here are Chrome's 25 dependencies as time of writing:

- Weak copyleft license: hunspell, Mozilla Interface to Java Plugins API, npapi, nspr, nss, Pthreads for win32
- Permissive license: bsdiff, bspatch, bzip2, dtoa, ICU, JSCRE, libpng, libjpeg, libxml, libxslt, modp_b64, V8 assembler, webkit, WTL, zlib

- “Public domain”: LZMA SDK, sqlite, tlslite

The absence of strong copyleft license in Chrome is logical because of inheritance: as explained in Appendix B, if Google would have used a GNU/GPL licensed program, Chromium would have been licensed under the GNU/GPL license, and Chrome as well: the GNU/GPL license prevent from being used in a proprietary program, whereas permissive licensed programs (like the 3-clause BSD license, see Appendix A) can be re-used anywhere, provided the author is cited. Because of the permissive nature of licenses protecting third-party software like WebKit or V8, Google could have made its browser a proprietary software. It could have also used Firefox’s layout engine Gecko (licensed under the MPL), or even Spidermonkey, its JavaScript engine.

Appendix C

What we talk about when we talk about cloud computing

From Sheth [2009] on the Official Google Enterprise Blog.

> Recently, McKinsey & Company published a study on cloud computing as part of a symposium for The Uptime Institute, an organization dedicated to supporting the enterprise data center industry. We share McKinsey's interest in helping the IT industry better understand cloud computing, so we'd like to join the conversation Appirio and others have started about the role of cloud computing for large enterprises.
>
> There's quite a bit of talk these days about corporations building a "private cloud" with concepts like virtualization, and there can be significant benefits to this approach. But those advantages are amplified greatly when customers use applications in the scalable datacenters provided by companies like Google, Amazon, Salesforce.com and soon, Microsoft. In this model, customers can leverage hardware infrastructure, distributed software infrastructure, and applications that are built for the cloud, and let us run it for them. This offers them much lower cost applications, and removes the IT maintenance burden that can cripple many organizations today. It also allows customers to deliver innovation to their end users much more rapidly.
>
> We thought we'd provide some insight into what we mean when we say cloud computing, and how its advantages in cost and innovation continue to attract hundreds of thousands of companies of all sizes – from 2nd Wind Exercise Equipment to Genentech. We created our cloud by building an optimized system from the ground up: starting with low-cost hardware, adding reliable software infrastructure that scales, offering innovative applications, and working every day to improve the whole system. While the McKinsey study only considered the hardware cost savings of the cloud, there is tremendous customer benefit in all of these areas.
>
> **Hardware infrastructure.** It starts with components. We serve tens of millions of users, so we've had to build infrastructure that scales and can run extremely efficiently to support that load. Consider three areas of data center design: server design, energy efficiency, and scale of operations.
>
> In the virtualization approach of private data centers, a company takes a server and subdivides it into many servers to increase efficiency. We do the opposite by taking a large set of low cost commodity systems and tying them together into one large supercomputer. We strip down our servers to the bare essentials, so that we're not paying for components that we don't need. For example, we produce servers without video graphics chips that aren't needed in this environment.
>
> Additionally, enterprise hardware components are designed to be very reliable, but they can never be 100% reliable, so enterprises spend a lot of time and money on maintenance.

In contrast, we expect the hardware to fail, and design for reliability in the software such that, when the hardware does fail, customers are just shifted to another server. This allows us to further lower the cost of our servers by using commodity parts and on-board storage. We also design the systems for easy repair such that, if a part fails, we can quickly bring the server back into service.

Traditionally, companies have focused on using large, highly reliable hardware to run databases and large backend systems, but there is a significant cost impact to that strategy. For example, a 4 CPU quad-core system with 600 GB of high end SCSI storage and 16GB of memory is 8 times more expensive than a system 1/4 its size with less expensive SATA storage. This is because the price of the components increase exponentially as the hardware gets larger and more reliable. By building the reliability into the software, we're able to use a much lower cost hardware platform but still maintain the same reliability to customers.

Beyond server design, we do everything possible to make our servers and data centers as efficient as possible from an energy and cooling perspective. Consider how we designed our data centers for energy efficiency. Power Usage Effectiveness (PUE) is an industry-standard metric for measuring the efficiency of a data center. We recently shared that the average PUE for our data centers is now better than the state-of-the-art 2011 data center PUE prediction by the EPA. In other words, we beat the EPA's best case estimates three years early, and we achieved this result without the use of exotic infrastructure solutions thought necessary in the EPA report. And we're doing that at every level of the stack: from server utilization to networking.

Finally, we operate at scale, and that drives economies of scale. Just by managing thousands of servers together and making them homogeneous, we're able to cut down on our administrative costs dramatically and pool resources of many types. This benefits end users by enabling us to offer low prices.

But, most importantly for our customers, we manage this entire infrastructure such that they don't have to. According to Gartner, a typical IT department spends 80% of their budget keeping the lights on, and this hampers their ability to drive change and growth in their business. The reality is that most businesses don't gain a competitive advantage from maintaining their own data centers. We take on that burden and make it our core business so that our customers don't have to.

Software Infrastructure. While most discussions of cloud computing and data center design take place at the hardware level, we offer a set of scalable services that customers would otherwise have to maintain themselves in a virtualization model. For example, if a company wanted to implement a typical three tier system in the cloud using virtualization, they would have to build, install, and maintain software to run the database, app server, and web server. This would require them to spend time and money to acquire the licenses, maintain system uptime, and implement patches.

In contrast, with a service like Google App Engine, customers get access to the same scalable application server and database that Google uses for its own applications. This means customers don't have to worry about purchasing, installing, maintaining, and scaling their own databases and app servers. All a customer has to do is deploy code, and we take care of the rest. You only pay for what you need, and, with App Engine's free quota, you often don't pay anything at all.

A great example of software infrastructure that scales is the recent online town hall meeting held by President Obama. The White House was able to instantly scale its database to support more than 100,000 questions and in excess of 3.5 million votes, without worrying about usage spikes that typically would be tough to manage. Because of the cloud, there was no need to provision extra servers to handle the increased demand or forecast demand ahead of time.

Applications. Beyond the underlying hardware and software design, what attracts many customers to the cloud is application outsourcing.

There is limited value to running an Exchange Server in a virtual machine in the cloud. That server was never designed for the cloud, so you don't get additional scale. You'd also need to continue to maintain and monitor the mail server yourself, so the labor savings are marginal. But with cloud-based applications like Gmail, we take care of all of the hassle for you. We keep the application up and running, and have designed it to scale easily. All of this provides an application that is roughly less than 1/3 the cost of a privately hosted mail system, has 100x the typical storage, and innovates much faster.

Innovation. While the cost advantages of cloud computing can be great, there's another advantage that in many ways is more important: the rapid pace of innovation. IT systems are typically slow to evolve. In the virtualization model, businesses still need to run packaged software and endure the associated burden. They only receive major feature enhancements every 2-3 years, and in the meantime they have to endure the monthly patch cycle and painful system-wide upgrades. In our model, we can deliver innovation quickly without IT admins needing to manage upgrades themselves. For example, with Google Apps, we delivered more than 60 new features over the last year with only optional admin intervention.

The era of delayed gratification is over – the Internet allows innovations to be delivered as a constant flow that incorporates user needs, offers faster cycles for IT, and enables integration with systems that were not previously possible. This makes major upgrades a thing of the past, and gives the customer greater and greater value for their money.

As companies weigh private data centers vs. scalable clouds, they should ask a simple question: can I find the same economics, ease of maintenance, and pace of innovation that is inherent in the cloud?

Appendix D

Creative Commons License

(A more easily readable version of this license is available here: http://creativecommons.org/licenses/by-sa/3.0/)

Creative Commons

Creative Commons Legal Code

Attribution-ShareAlike 2.0

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

1. "Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

2. "Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

3. "Licensor" means the individual or entity that offers the Work under the terms of this License.

4. "Original Author" means the individual or entity who created the Work.

5. "Work" means the copyrightable work of authorship offered under the terms of this License.

6. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

7. "License Elements" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

1. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;

2. to create and reproduce Derivative Works;

3. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

4. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

5.

For the avoidance of doubt, where the work is a musical composition:

1. Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

2. Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights society or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

6. Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions.The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

1. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any reference to such Licensor or the Original Author, as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any reference to such Licensor or the Original Author, as requested.

2. You may distribute, publicly display, publicly perform, or publicly digitally perform a Derivative Work only under the terms of this License, a later version of this License with the same License Elements as this License, or a Creative Commons iCommons license that contains the same License Elements as

this License (e.g. Attribution-ShareAlike 2.0 Japan). You must include a copy of, or the Uniform Resource Identifier for, this License or other license specified in the previous sentence with every copy or phonorecord of each Derivative Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Derivative Works that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder, and You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Derivative Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Derivative Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Derivative Work itself to be made subject to the terms of this License.

3. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and give the Original Author credit reasonable to the medium or means You are utilizing by conveying the name (or pseudonym if applicable) of the Original Author if supplied; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE MATERIALS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

1. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

2. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

1. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

2. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You

under this License.

3. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

4. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

5. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, neither party will use the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time.

Creative Commons may be contacted at http://creativecommons.org/.